Computers & Education 59 (2012) 974–988

Contents lists available at SciVerse ScienceDirect

Computers & Education

journal homepage: www.elsevier.com/locate/compedu

A LEGO Mindstorms NXT approach for teaching at Data Acquisition, Control Systems Engineering and Real-Time Systems undergraduate courses

A. Cruz-Martín, J.A. Fernández-Madrigal, C. Galindo*, J. González-Jiménez, C. Stockmans-Daou, J.L. Blanco-Claraco

System Engineering and Automation Department, University of Málaga, ETSI Informatica, Campus Teatinos, 29071 Malaga, Spain

ARTICLE INFO

Article history:
Received 29 October 2011
Received in revised form
27 March 2012
Accepted 31 March 2012

Keywords:
Applications in subject areas
Improving classroom teaching
Interactive learning environments
Post-secondary education
Teaching/learning strategies

ABSTRACT

LEGO Mindstorms NXT robots are being increasingly used in undergraduate courses, mostly in robotics-related subjects. But other engineering topics, like the ones found in data acquisition, control and real-time subjects, also have difficult concepts that can be well understood only with good lab exercises. Such exercises require physical educational tools that should be low cost, easy to configure and use, multi-purpose and motivational for the students, being all of this hard to achieve with a single device. The "classical" solution has been to acquire specific commercial kits for each subject, or even topic, usually proprietary and expensive. Our work extends the already existing alternative of using the LEGO Mind-storms NXT robots as a training platform, but not by imitating the same approach of commercial kits (e.g., to isolate some part of the robot for teaching a particular topic); we rather aim at accomplishing all the mentioned requirements *simultaneously*. For that purpose, we have used only one out-of-the-box, complete robot configuration, to be shared among different subjects without hardware/software/firm-ware modifications. This has reduced significantly the effort of a group of professors when preparing exercises, and encouraged the reuse of their work among several topics and subjects. Also, we have collected a number of surveys on students and the professors' experiences. In this paper we describe our approach and present in detail the results, which assess the higher motivational adequacy of using a complete robot in these subjects and also the real fulfillment of the other requirements along several academic years.

1. Introduction

In a world where technology leads the economic and social development of the countries, the significance of a good academic engineering training must be considered as a core aspect in every educational system. Such engineering training must cover both theoretical aspects as well as practical applications that show the students how to relate the abstract knowledge they learn in the lecture sessions with real world problems and their difficulties.

A cornerstone, or at least an important part, of the curriculum in many engineering degrees (Computer Science, Electrical or Electronics, Mechatronics, and so on) are the Data Acquisition, Control Systems Engineering and Real-Time subjects. Teaching this sort of subjects reveals a major difficulty: a wide and complex theoretical base is needed in controlled or real-time systems, but this reduces the time devoted to practical applications, that in turn are essential to understand what has been explained in the lecture sessions.

This chicken-and-egg issue has been ordinarily solved by preparing a set of laboratory sessions using proprietary commercial training devices, like Lucas-Nülle (2011), Feedback plc (2011). However, these educational solutions, though robust and efficient, are also usually expensive and, in some situations, not versatile enough to cope with the variety of problems that the teacher wants to illustrate. If we change the point of view, and focus on inexpensive versatile systems, we can find in the market some interesting solutions, such as Arduino microcontrollers boards (Arduino, 2011), but they also have drawbacks: a lot of extra components must be added to get a functional control

* Corresponding author.
E-mail addresses: anacm@ctima.uma.es (A. Cruz-Martín), jafma@ctima.uma.es (J.A. Fernández-Madrigal), cipriano@ctima.uma.es (C. Galindo), jgonzalez@ctima.uma.es (J. González-Jiménez), jlblanco@ctima.uma.es (J.L. Blanco-Claraco).

0360-1315/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compedu.2012.03.026

or real-time system, and their technical requirements could force the students to concentrate on programming instead of on the control and real-time concepts.

An innovative solution is the use of robotic tools for these subjects. Robotics in education is an expansive trend, and at present, more and more educational institutions, ranging from elementary school to college, are including robotics in their curricula (the interested reader may consult (Miller, Nourbakshs, & Siegwart, 2008) for a deeper review of the state of the art in educational robotics). Many educational robotic platforms spring nowadays, giving the teachers the chance to select the most suitable one for their teaching purposes. For example, (Oliver, Toledo, & Valderrama, 2010; Summet et al., 2009; Touretzky & Tira-Thompson, 2010) in Computer Science and Computer Engineering degrees, or ECEBot (Maher, Becker, Sharpe, Peterson, & Towle, 2005) and BoeBot (Harbour & Hummel, 2010) in engineering courses. Regarding remote educational tools, there are several solutions, like Khamis, Rodríguez, and Salichs (2003), related to indoor robotics. Unfortunately, these robotics platforms are mostly employed in robotics-related subjects.

We show in this paper how a particular robotic platform can provide simultaneously the versatility, low cost and a suitable level of user requirements of other, non-robotics subjects: the LEGO Mindstorms robots (LEGO Mindstorms, 2011). In particular, we present the work developed by a group of professors at Málaga University involved in Data Acquisition, Control Systems Engineering and Real-Time Systems subjects during the 2008–2009 and 2009–2010 academic years. We have designed, developed and applied in real classes during those periods a set of practical lab exercises implemented on a complete configuration of the LEGO Mindstorms NXT robot. Thanks to these exercises, we have detected that students from different disciplines are able to translate the theoretical aspects of the lectures to a real robotic platform, checking how the theory actually works, facing the difficulties that physical systems have, and improving their involvement in their learning process. We claim this supposes a significant improvement of the quality of our teaching. The particular goals we report in this paper are:

– To briefly describe a varied set of lab exercises related to Data Acquisition, Control Engineering and Real-Time systems courses that we have developed with the NXT robot. They are designed for undergraduate levels in different academic programs, without robot hardware, software or firmware reconfiguration between courses or exercises. These exercises have been used in different academic subjects during several years, they are still used, and they have been extended to other subjects after the work presented here was finished.
– To analyze deeply the results we have obtained in a two-fold manner: firstly, we apply an objective statistical point of view to the questionnaires our students have completed, so we have measured the benefits and difficulties they have detected, which has allowed us to modify our original teaching strategies afterward; secondly, after a subjective analysis of the advantages and drawbacks we have met while working with the robot along two academic courses, we show that an out-of-the-box NXT physical configuration is really well suited for teaching at undergraduate levels, and accomplishes the economic and versatility requirements previously mentioned in this section, though some logistics must be considered in order to get the best from the robots.

The paper is organized as follows. Section 2 serves as a brief survey on the applications of the LEGO Mindstorms robots in engineering education. Section 3 concisely displays the off-the-shelf hardware configuration of the robot we have chosen for our proposal, whereas Section 4 explains the subjects at Málaga University involved in this innovative teaching approach. Section 5 summarizes the different laboratory exercises we have prepared. The obtained results, regarding both to the point of view of the students and the conclusions drawn from the professors, have really our main interest in this paper, and thus are presented in detail in Section 6. Finally, Section 7 is devoted to the general conclusions of our work and the future tasks we would like to complete.

2. Related work

LEGO-based solutions have already been reported in Computer Science/Computer Engineering related areas like programming, AI, embedded systems or mechatronics (Fiorini, 2005; LEGO Engineering, 2011; Williams, 2003). For our purposes, we have classified the NXT-based educative literature under three possible axes: RCX-based works (the RCS was the first of the LEGO Mindstorms robots, being superseded by new models), NXT non-control-related engineering works, and, finally, NXT data acquisition, control engineering/real-time oriented works, where our approach lays.

There are several innovative educational solutions built on the ancestor of the LEGO NXT, the RCX brick (LEGO Mindstorms, 2011), some of them aimed to children (age 7–14), like Lund and Pagliarini (2000), Nagchaudhuri, Singh, Kaur, and George (2002), Lau, Tan, Erwin, and Petrovic (1999), Aslam, Cao, and Rostamzadeh (2008). In a general way, and for undergraduate levels, in Klassner and Anderson (2002) the authors show how RCX can be used in order to meet the ACM Computing Curriculum 2001. Some authors have used RCX as a means for retaining freshmen at engineering courses: Mota (2007), Pomalaza-Ráez and Henry Groff (2003). Others worked with RCX particularly in embedded systems courses, like Han Kim and Wook Jeon (2009). RCX is also employed in Cliburn (2006) for developing programming skills in undergraduate Computer Science students; that paper is interesting since most of the literature about RCX/NXT in the teaching realm focuses on the advantages of using this platform, but there are little reference on cases where including the robot as a part of a subject does not meet the initial improvement expectations (may be, only Fagin and Merkle (2003), where some troubles on an Ada based introductory programming course for freshmen have been detected). Several authors have used RCX for robotics subjects. For example, Benedetelli, Casini, Garulli, Giannitrapani, and Vicino (2009) present the RCX as an experimental setup for evaluating multiagent systems in undergraduate robotics courses. Behavioral robotics knowledge is reinforced in a course targeted at non-freshmen students with weak programming experiences in Computer Science, Engineering and Psychology programs (Gage & Murphy, 2003); that work also presents an interesting description and scheduling of the lab exercises they propose. The work by Greenwald, Artz, Mehta, and Shirmohammadi (2006) shows a RCX used to perform a particle-filter based robot localization solver. In Galvan, Botturi, Castellani, and Fiorini (2006) the goal is to mix theoretical lectures with laboratory lectures where a 3 DOF robotic manipulator is built, and then its kinematic behavior is designed. Concerning the subjects of our interest, the work by Gawthrop and McGookin (2009) shows how to use the RCX for different teaching activities related to control systems, so this is connected to the goals we pursue in our approach. This use is focused on four aspects: demonstrators for lectures (which we do not use), laboratory exercises, undergraduate projects, and research, which are the fields we try to cover with the NXT. Apart from the natural differences due to timeline (they used RCX since NXT was not yet released), there are some subtle variations between both approaches: we do not include competition among students as a part of the courses, and we have carried out an evaluation process in the subjects in order to obtain quantitative and qualitative records that we use in order to improve our teaching. The work by Wang, LaCombe, and Rogers (2004) presents a set of laboratory experiments with RCX that can be developed for different aspects of a variety of courses at undergraduate level. One of these aspects is related to feedback control: they propose different practical exercises, some of them from the literature, to be solved by means of bang–bang or proportional controllers (our approach here covers the line follower, both using on-off and PID controllers) That paper concludes with an interesting study of the difficulties that can be found when LEGO RCX robots are adapted as a laboratory tool.

Regarding the NXT platform, some teaching works use it for subjects that do not strictly refer to the control systems engineering or real-time subjects. So, in Fernández Panadero, Villena Román, and Delgado Kloos (2010), the NXT is used for programming, robotics, and AI subjects. This robotic platform is also used for teaching advanced software development (Lew, Horton, & Sherrif, 2010), mechatronics (Gómez-de-Gabriel, Mandow, Fernández-Lozano & García-Cerezo, 2011; Jaksic & Spencer, 2007), electrical engineering (Ranganathan, Schultz, & Mardani, 2008) and, of course, robotics: in Klassner and McNally (2007), they propose the NXT to be used in the AI curriculum for undergraduate students dealing with robotic mapping, localization or path planning issues. The work by Cuéllar and Pegalajar (2011) uses LEGO NXT for teaching AI agents to third-year undergraduates in Computer Engineering as well as the benefits and drawbacks of their solution. In Hamada and Sato (2011) some aspects of Theory of Computation related to Finite State Automates are shown by means of a FSA Simulator and a LEGO NXT robot. In Behrens et al. (2010), the NXT and MATLAB (Mathworks, 2011b) are used for dealing more easily with mathematical methods, DSP and programming for freshmen courses. In Sharad (2007), the NXT is used for explaining embedded systems aspects to engineering freshmen and sophomore students. An interesting approach consists of recruiting engineering undergraduates as paid mentors for elementary school NXT-based courses intended to encourage students to pursue STEM careers (Karp, Gale, Lowe, Medina, & Beutlich, 2010); the mentoring job also helps to improve retention rates among the engineering freshmen.

Finally, in the control engineering and real-time systems area, we found several approaches based on the NXT, but not focused on the simultaneous achievement of the requirements explained in the introduction or completely implemented in several real courses as ours. For example, in Grega and Pilat (2008), real-time techniques are proposed for controlling the motors of the NXT (it is not clear if this interesting application has been carried out by students in a real subject). Valera, Vallés, Fernández, and Albertos (2009) presents a set of practical activities related to controller design, as well as some other high level tasks like sumo competitions, though it seems they have not been applied to real courses either. The work by Kim (2011) shows a more similar idea to the one presented in our work: the NXT is used for control engineering lab sessions (not for real-time or data acquisition systems), and he also includes a final assessment of the student's obtained results. The difference between both approaches relies on which engineering issues are covered and on the motivational value and assessment of the solution. Kim focuses on modeling, PID control, and state-space control of one the motors of the NXT. Our solution does not cover advanced concepts like state-space, but deals with basic control topics such as stability and steady-state error and also with data acquisition from a diversity of sensors. Furthermore, we use the whole robot (not only the motors, but a physical robot that also uses its exteroceptive sensors) to perform high level control tasks like line following (in the context of automatic control). In our opinion, this is more appealing for the students in order to get involved with the practical lab exercises. From a logistic point of view, we work with the original firmware of the robot, without replacing it with another software; this offers the chance to share the robots with any other subjects that also uses the initial software and configuration of the NXT.

3. The common LEGO Mindstorms NXT robot configuration

We have selected the LEGO Mindstorms NXT (LEGO Mindstorms, 2011) for engineering training mainly due to budget reasons, ease and versatility of programming, variety of sensors, and robustness. In the following, we describe very briefly the physical structure and the programming platform we have chosen for implementing our approach as a common training platform for different topics.

The physical configuration we have used is one of the proposed by LEGO in the kit building guide, and is depicted in Fig. 1. Due to that reason, we do not enter into much detail here. Three sensors are connected: ultrasonic (on top of the brick), sound (on the right side) and light (at the bottom). Two motors are connected, both of them situated symmetrically at each side of the robot; a caster wheel (not actuated) is mounted on the rear side.

Fig. 1. NXT physical configuration used in all the courses.

The brick can be programmed using basic commands written in the brick itself, which only permits simple programs, or by using a more complex programming language that requires a PC for the corresponding IDE (Integrated Development Environment) to be deployed; in this latter case, programs are transferred from the computer to the brick via an USB or Bluetooth connection. Related to this high level programming issues, LEGO provides NXT-G, a LabVIEW (NI LABVIEW, 2011) based language, but we have chosen instead NXC (Hansen, 2007; NBC/NXC, 2011) for the majority of our exercises, a free C-like programming language for NXT. The choice of this language has been supported by several benefits: it runs on the original firmware of the robot without modification, most of our engineering students are already used to this language, C is an important programming language in real-time embedded systems and also shares the basic syntax constructions of many other languages, it has a proper documentation, and it works with the free and open-source programming environment Bricx Command Center (Brick, 2011) which alleviates the programming, maintenance, compilation and download tasks. Anyway, a lot of different programming languages have been developed for the NXT (LEGO NXT and Control System Teaching, 2010), that configures a wide and lively community of hardware and software developers.

4. Subjects involved in the experience

The subjects involved in our work belong to different Computer Science Engineering and Telecommunication Engineering undergraduate programs at Málaga University. A summary of their curricular situation is shown in Table 1.

These characteristics depend upon the Spanish university system, so some explanations must be done. First, since our work was developed during the academic years 2008–2009 and 2009–2010, the number of credits are counted following the pre-European Higher Education Area normative in Spain, so one credit was equivalent to ten hours of class, and therefore it should not be mixed up with the new European Credit Transfer System, which applies in the Spanish engineering academic programs from the 2010 to 2011 course. Second, the difference between the terms “Technical Engineering” and “Engineering” lays on the duration of the programs: in the Spanish *pre-Bolonia* scenario, a Technical Engineering was a three years program, whereas an Engineering took five years to be finished.

The context of these subjects in their respective programs is as follows.

Control Systems Laboratory (CSL) is the second experience of the students with Control Engineering. All of them should have followed a previous subject focused on the theoretical fundamentals; then in CSL they are exposed to practice issues, some of them related to real physical systems, like the NXT robot. The homogeneity of the students is therefore high, since both subjects are mandatory and sequential in the program. CSL classes include a theoretical part, which includes the explanation of the guideline of each LEGO exercise and a final open talk to discuss its solution, and in the meantime the students work the practical exercise in small groups (2 people), requesting for the teacher’s help if needed.

Computer Controlled Systems (CCS) is also a second experience with control engineering, but in a different program (Computer Science Engineering). The students come from a mandatory subject in whom they have studied both theoretically and with Matlab the fundamentals of dynamic systems and automatic control, but focused in continuous time and external representations (transfer functions). In CCS they review the main concepts of continuous systems and then enrich that knowledge with discrete and state-space control. This non-mandatory subject is followed by students with homogeneous skills and knowledge. The practical exercises cover the last 25% of the subject (about 15 h).

Control and Data Acquisition Systems (CDAS) is a very introductory subject intended for students of the second/third year of several engineering programs to have their first experience with control engineering and data acquisition. The subject is structured in a 50% of the class time (about 45 h) dedicated to theoretical explanations in the lab plus another 50% for developing the concepts through some practical problem that the students must solve in small groups with the aid of the professor. Each day along the term is split into that 50/50 scheme. One third of the term is devoted to dynamical systems, another third to data acquisition systems and the final part to direct control systems (LTI, continuous). The number of students is reasonable for the situation of this non-mandatory subject, but their diversity is high, since they are coursing different programs and have had different levels of exposure to its topics previously.

Finally, Real-Time Systems (RTS) is a mandatory, very crowded subject that is coursed in the last year of the Technical Computer Science program. The students are, therefore, homogeneous, and have had no previous exposure to these topics (except for microcontrollers and programming issues). The main problem with this subject is the number of students; the lab exercises occupy the last third of the term. In this case the students use the robot not for the exercises, but for a set of voluntary works that help them to improve their marks, with a duration of about 6 h, placed at a point where they have finished the explanations of the main theoretical concepts about small-scale real-time systems.

In summary, the students of the different subjects are homogeneous only in their related engineering careers, and they are sophomore or higher. There is a relevant difference among some of the courses: the number of students in the class. RTS has a high number of enrolled students, so different lab shifts were scheduled in order to attend them in the same conditions as students from subjects with less number of students. Our laboratory has 24 stations with a PC. We have used 10 LEGO NXT robots, and the students have worked in groups of 2 or 3 persons in all the subjects. The introductory practice explained in the next section is shared by these subjects it is aimed to introduce the students to the LEGO Mindstorms NXT platform, which is new for all of them. All the practices are designed for autonomous work of the students, i.e., they self-learn to use the theoretical concepts explained previously onto the physical robot. The solution to each practical

Table 1
Description of the curricular situation of the courses involved in our work. Exhaustive information on each BS curriculum can be found in www.infouma.uma.es/estudios.

Subject	Credits	Academic program	Year in academic program	Semester	No. of students (2009–2010)
Control Systems Laboratory (CSS)	4.5	Technical Telecommunications Eng.	3 Mandatory	1 (Sept.–Feb.)	43
Computer Controlled Systems (CCS)	6	Computer Science Engineering	Optional	1 (Sept.–Feb.)	2
Control and Data Acquisition Systems (CDAS)	9	Computer Science Engineering, Technical Computer Science Eng., Technical Telecommunications Eng.	Optional	2 (Feb.–Jun.)	9
Real-Time Systems (RTS)	4.5	Technical Computer Science Eng.	3 Mandatory	1 (Sept.–Feb.)	218

exercise is only provided schematically, in some cases, and at the end of the lab class time: the main feedback with the students consists rather of individualized assistance when they get stuck, can clearly improve their proposals or can be benefit from a more abstract view of the topics at hand and of their inter-relationships.

5. Summary of the practical exercises

The exercises presented in this section can be used by different courses in different ways, though for each kind of exercise we give a hint of which subjects fit better. For example, the introductory exercise can be used as a single lab session, or can be spread among several lab sessions as a complementary part in order to get the students prepared for working with the robot. In other cases, some exercises have been sequenced together so they cover the time assigned to a laboratory session.

We do not enter in much detail in the description of these exercises, since the focus of the paper is rather to analyze the results obtained after their implementation. An example of the proposed exercises can be found in CV-UMA (2010).

5.1. Introductory exercise

We have prepared an introductory exercise that allows the students to face the main programming aspects they will have to deal with when using the robots under the NXC/BricxCC environment. The exercise includes the use of constants, variables, basic control structures, access to sensors and actuators, display output and file operations (in order to recover data logged in the NXT robot and pass them to a more powerful analysis software, like Matlab).

Since this exercise covers the robot basic programming issues, it is well suited for all the subjects we cope with.

Furthermore, we provide our students with the NXTpie wiki (NXTPIE, 2010) where they can find extended information about tools, material, languages and application related to the NXT robot.

5.2. Modeling exercises

In these exercises, the students must use the Laplace transform in order to model and simulate the LEGO NXT robot motion system, which reinforces the following topics: modeling based on linear ordinary differential equations (ODEs) of rotational, translational and electrical systems, Laplace transform of ODEs, solving systems of equations, simulation of high-order LTI systems, on-line acquisition of data from an embedded system, and rotational position sensors (i.e., encoders). The model in this exercise relates an input motor power value with the traveled distance of the robot on the floor, along a straight line, assuming the physical configuration previously shown. In order to make things easier, the robot is subdivided into different subsystems (DC motors, rotational subsystem, etc), and several assumptions that simplify the calculations have been taken into account. This subdivision generates at the end eight linear, time-invariant differential equations that must be solved in order to get the final model. Finally, the students use this model to build up a Simulink simulation and check how the theoretical model fits the behavior of the real physical robot.

This exercise fits well in CDAS and CSL, since CCS is focused on discrete and space-state models.

5.3. Control engineering exercises

The set of exercises related to control engineering covers the following theoretical topics, explained previously in the corresponding subject:

- Open loop vs. closed loop control. The students are requested to control the robot movement in order the motors to rotate the wheels up to some given encoder value; first via an open loop controller, and then using a feedback controller. The students can observe how both controllers behave, and also their fundamental differences, especially when disturbances appear.
- Stability. The robot is used as an inverted pendulum (i.e., the main brick is a mass that rotates around the wheels' axle, and it must be kept upright manually). From a simplified model of the robot as a rigid body, the students apply linearization techniques to get a second order transfer function for the robot. Then, a Routh–Hurwitz table is built in order to set the range of K for stability. Finally, a closed loop proportional controller that keeps the mass of the robot upright automatically (against gravity) is designed and implemented.
- Bang–bang control: A bang–bang controller is implemented for the robot to follow a concrete light value when it is moving on an elliptic gray-shaded floor surface. The students have to get graphical representations of different executions with different values of power and hysteresis, in order to see the benefits and drawbacks of this kind of controllers.
- PID control: a line following task is implemented. This line follower is based on the work of Sluka (2009), though we have added a light-calibration phase in order to avoid differences in the performance of the controller due to ambient light issues (some problems related to light have been reported in Han Kim and Wook Jeon (2009) and Cliburn (2006).

Again, this set of exercises are intended for CSL and CDAS courses.

5.4. Data acquisition exercises

We have prepared a lab session devoted to data acquisition from the sensors provided by the LEGO NXT. Obviously, this is aimed to CDAS, but can also be used in other subjects as an introduction to the sensory part of any controlled system. The students, by launching different programs that log data in the flash memory of the robot, transmitting the data to the PC, and then analyzing such data in MATLAB, study the following topics of the CDAS subject:

- ○ Ultrasonic sensor. The lower and upper limit of the measures the sonar offers are recorded in order to set the work range of this sensor. After that, a set of sonar measurements are taken, so the average sensitivity of the sensor can be estimated.

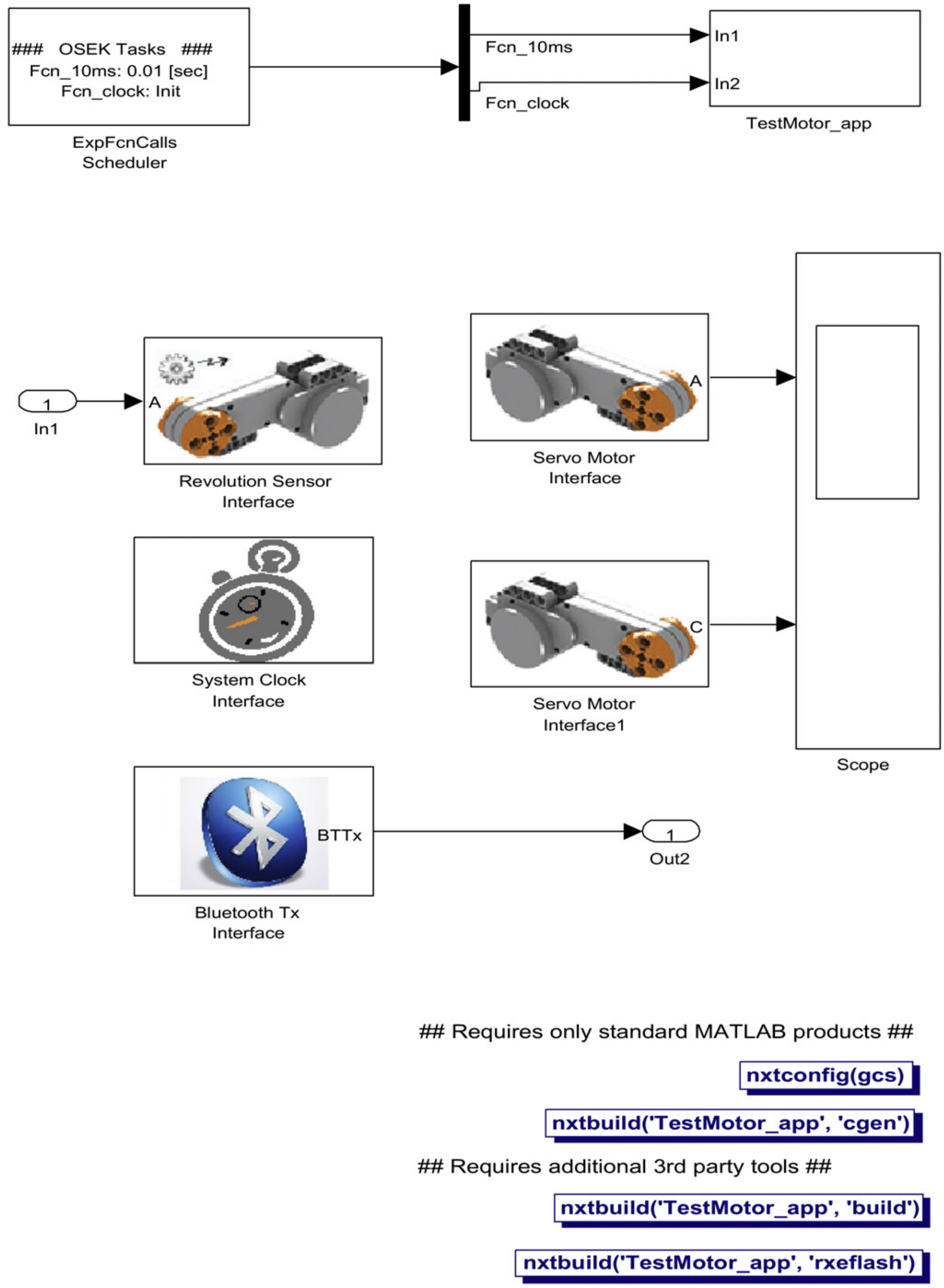


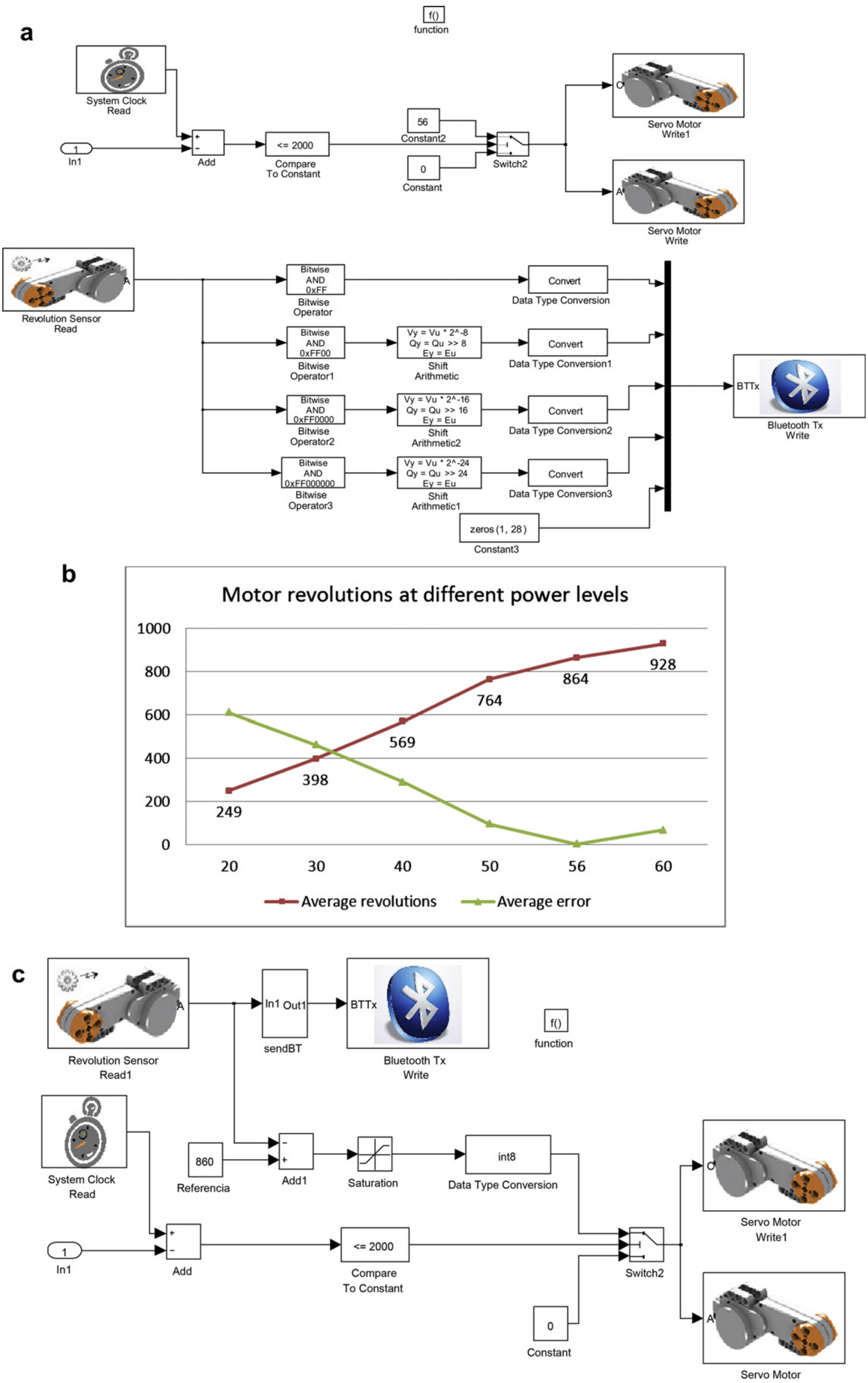
Fig. 2. ECRobot components involved in the experiences.

- ○ Motor encoder. The repeatability of this sensor is obtained by analyzing the histogram, average and deviation of the encoder of the wheels recorded after a loop of 200 iterations with the wheels of the robot unloaded and synchronized.
- ○ Finally, the resolution of the light sensor is calculated by means of a loop that records the light measurements taken while the robot moves on a surface linearly shaded from black to white.

5.5. *Real-Time exercises*

The goal of RTS practices, which are aimed specifically to the RTS subject, was to allow students to face the actual hurdles one can find while designing software intended to interact in real-time with some pieces of hardware. We designed three practical exercises of increasing complexity, thus the experience acquired by the students in the first sessions helped them to achieve the goals of subsequent ones:

- ○ The first practical session has a three-fold goal. First of all, to introduce the NXC programing language to the students in case that the introductory exercise described in Section 5.1 takes too long for including it into the course. Secondly, this exercise includes a guided demonstration of the *Bricx Command Center*, the aforementioned integrated development environment (IDE) for the NXC language.


a
f()
function
System Clock
Read
1
In1
Add
<= 2000
Compare
To Constant
56
Constant2
0
Constant
Switch2
Servo Motor
Write1
Servo Motor
Write
Revolution Sensor
Read
Bitwise
AND
0xFF
Bitwise
Operator
Bitwise
AND
0xFF00
Bitwise
Operator1
Bitwise
AND
0xFF0000
Bitwise
Operator2
Bitwise
AND
0xFF000000
Bitwise
Operator3
Vy = Vu * 2^-8
Qy = Qu >> 8
Ey = Eu
Shift
Arithmetic
Vy = Vu * 2^-16
Qy = Qu >> 16
Ey = Eu
Shift
Arithmetic2
Vy = Vu * 2^-24
Qy = Qu >> 24
Ey = Eu
Shift
Arithmetic1
Convert
Data Type Conversion
Convert
Data Type Conversion1
Convert
Data Type Conversion2
Convert
Data Type Conversion3
zeros (1, 28)
Constant3
BTTx
Bluetooth Tx
Write
b
Motor revolutions at different power levels
1000
800
600
400
200
0
20
30
40
50
56
60
249
398
569
764
864
928
Average revolutions
Average error
c
Revolution Sensor
Read1
In1 Out1
sendBT
BTTx
Bluetooth Tx
Write
f()
function
System Clock
Read
860
Referencia
Add1
Saturation
int8
Data Type Conversion
1
In1
Add
<= 2000
Compare
To Constant
Switch2
0
Constant
Servo Motor
Write1
Servo Motor
Write

Finally, the third goal of this exercise is to reveal some of the basic difficulties of real-time robotic systems, such as the random component of each robot movement or the unpredictability of motor actions and delays in software loops. To achieve all these goals, students are instructed to employ the *Brick Command Center* to write an NXC program that sets both robot motors to their maximum speed forward. Then, a software loop must watch for the odometry data and command the motors to stop when a predefined number of ticks are reached. After the robot stops, the traveled distance is measured by hand in order to provide a gross estimation of the tick-to-centimetre odometry conversion factor. This parameter is a characteristic of the robot and could be used in subsequent exercises. The experience is repeated several times and results averaged to compensate the random component in the robot motion. Then, the exercise proposes to introduce a series of delays into the software loop and watch for the expected encoder count, simulating the different latencies at which a program could expect to obtain the CPU running time in a shared-time operative system. The effects of those delays in the granularity of the control one can have on the expected robot path is clearly revealed.

- The second exercise aims at developing two different skills in students: (i) text manipulation in NXC and showing results on the robot display, and (ii) working out odometry-based estimations of traversed distances with real-time requirements. The students are instructed to write an NXC program to make the robot rotate on its own during a predefined period of time by setting both motors at the same speed but in opposite directions. Then, they have to find the proper velocity value to make the robot complete one exact turn, after which the encoder readings are used to compute the distance traversed by one wheel and that value printed in the display after some formating. This result is finally compared with a theoretical prediction in order to insist on the idea that robot actions are never exact and always contain some degree of inaccuracy, and that real-time requirements play an unavoidable role in uncertain systems.
- Finally, the third exercise involves a real closed loop feedback control system. The goal is to employ the light sensor of the NXT robot to design an on/off-like controller like the one explained in the Control Engineering Exercises section, which obviously is a typical example of real-time system. Apart from realizing of the effects of the robot response time in the quality of the path-tracking operation, the students must verify by themselves how easy is to implement a closed loop feedback system with this robotic platform (remember that these students have not to be enrolled in the Control Engineering courses).

5.6. *Computer Controlled Systems exercise*

The following exercise has been carried out in the Computer Controlled Systems (CCS) subject. Because of the reduced number of students enrolled in CCS during the period 2009–2010 (only four), the exercise considered consisted of the study of the toolkit Embedded Coder Robot for LEGO Mindstorms NXT (ECRobot NXT) (Matlab Central, 2011; Mathworks, 2011a). This is a programming environment that does not need the NXC language. The students were asked to develop a research exercise as a single group, using the ECRobot NXT, to study a Control System topic left up to them. In the following we describe this practical experience in a little more detail than the previous ones, due to its interest as a student-devised exercise.

The ECRobot NXT toolkit permits the integration of the Matlab/Simulink environment with the NXT robot. CCS students have intensively used the Matlab/Simulink software in previous courses and thus they can exploit all the capabilities that this software offers in the field of Control Engineering, i.e., the Control System Toolbox.

The research exercise chosen by the students was stating the difference between open loop and closed loop control systems which is one of the introductory topics of the course. They proposed the position control of the LEGO motors including a computer running the ECRobot toolkit as system controller. The ECRobot toolkit is in charge of managing the reference signal, the generated control signal and the system output, as well as the bidirectional communication to the LEGO robot via Bluetooth.

The implementation using the ECRobot toolkit requires a high level description and connection of the elements involved in the experience as shown in Fig. 2. In both configurations we consider two LEGO motors, revolution sensors to measure wheels' position, and the Bluetooth communication interface.

The objective of the conducted tests was the control of the LEGO motors to go forward 50 cm, which corresponds approximately to 860 ticks of the revolution sensors, under both control configurations.

The implementation of the open loop control is depicted in Fig. 3a. Given that there is no feedback, a reference signal had to be empirically obtained. Fig. 3b shows the revolutions given by the motors during two seconds at different power levels. A value of 56 (out of 100) is considered as the reference signal for the open loop control.

For the closed loop control system we have feedback information about the motors' position and thus the reference signal (in this case 860 revolutions) is continuously compared to the actual revolutions (see Fig. 3c). The control signal is generated through a saturation block to properly limit the motors' input.

With this exercise the students gained a practical insight into the difference between open and closed loop controlled systems which motivates the rest of the theoretical concepts of the course, like the need of appropriate controllers for physical systems.

6. Evaluation and results

This section is devoted to the detailed analysis of the obtained results for the last year of the academic experience, when some adjustments were made from the implementation of the first courses. We consider that this is the central contribution of this paper, thus we explain these results more in depth than the previously described lab exercises.

The analysis has a two-fold focus: the effects on the teaching aspects of each subject, and how the NXT has behaved during the development of our work from the point of view of the professors. We present here both quantitative and qualitative results, since the anonymous questionnaires the students have fulfilled included both fixed grading and free expression questions, in order to obtain both

Fig. 3. a) Implementation of the open loop configuration. b) Measured motor revolutions at different power levels. Notice that a motor input of 56 (out of 100) approximately achieves the desired output. c) Implementation of the closed loop configuration. The measured motor's position (feedback information) is used to generate the control signal.

statistical data as well as a more personal vision of the subject. We also present some quantitative results of officially (and anonymously) polling the students about some of the subjects involved in this work. These polls are intended as a quality improvement mechanism in our University. They are completed with some figures on the evolution of the scores of students before and after the LEGO robots were used.

6.1. Learning results: the student's point of view

The teaching results obtained by introducing the LEGO Mindstorms NXT in our subjects have been promising. From the student's point of view, they feel satisfied and highly motivated by using a complete robot, not suffering any significant inconvenience using it. From our teaching point of view, this two years' experience has been very positive, though we have also detected that more work has to be done for filling the gap between some concepts explained in theory classes and their practical application in lab sessions. We have also noticed that, actually, the use of a complete robot can be *too motivational*: students sometimes miss the point and do not focus as deeply as we wish on the data acquisition, control systems engineering or real-time systems aspects of the laboratory practical exercises, because they see the robot as a toy or want to address robotics issues.

Due to organizational reasons, we have not been able to implement, for a more accurate psychological comparison of the learning results, a control group of students that follows the subject without using the robots in the same academic year as we conducted our educational experience, but have retrieved some official polls about three of the subjects that are of interest and also compared the scores of students in three of the subjects before and after the robots were used. Fig. 4 shows the results for RTS, CDAS and CSL in three questions of the polls that are relevant for this work, both a course before and the same course using the LEGO Mindstorms NXT. The questions are: Q1) Does the professor use didactic resources to facilitate learning?, Q2) Does the professor provides practical examples for the theoretical concepts of the subject?, and Q3) Are you motivated by the professor in this subject? These are very general and basic questions, but we have found a consistent, slightly increase in the expectation after using the robots (only Q2 in CDAS and Q3 in CSL, i.e., 2 out of 9 figures, show a slight decrease in the results). All in all, this indicates that there is room for improvement in this approach during future academic years. This is confirmed by the results shown in Fig. 5: there we have compared the scores of the students in the main exams of two courses: the dark gray histograms are the relative frequencies of the marks obtained one academic course before our robot experiences, i.e., when no robots were employed in the subjects; the light gray histograms show the same for the first course were the robots were used. The results indicate a similar behavior to the one of the polls previously described: a slight improvement in scores can be observed, with a greater presence of better scores after the experience. We believe that this slight trend toward higher marks comes from a higher motivation of the students, although it is not substantiated in a more drastic change due to the complexity of the subjects.

In general, the work with the NXT in these subjects has aroused further interest in this robotic platform: currently there are six students preparing their BSc theses with these robots, covering aspects as diverse as integrating the NXT with wider robotic development application systems, environment mapping, or swarm robotics.

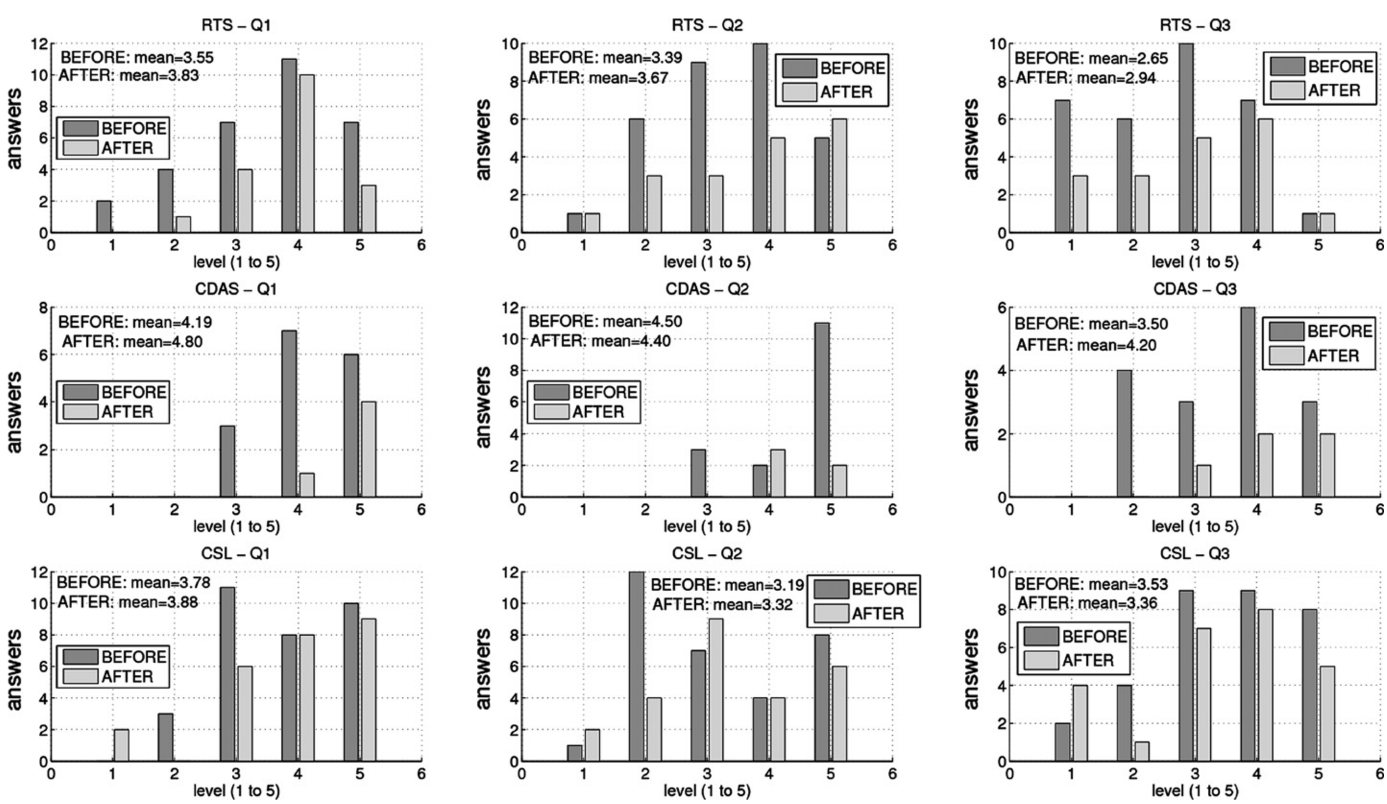


Fig. 4. Results of the official polls conducted in academic years where the robots were and were not used. Three questions are shown in the figures: Q1) Does the professor use didactic resources to facilitate learning?, Q2) Does the professor provides practical examples for the theoretical concepts of the subject?, and Q3) Are you motivated by the professor in this subject?

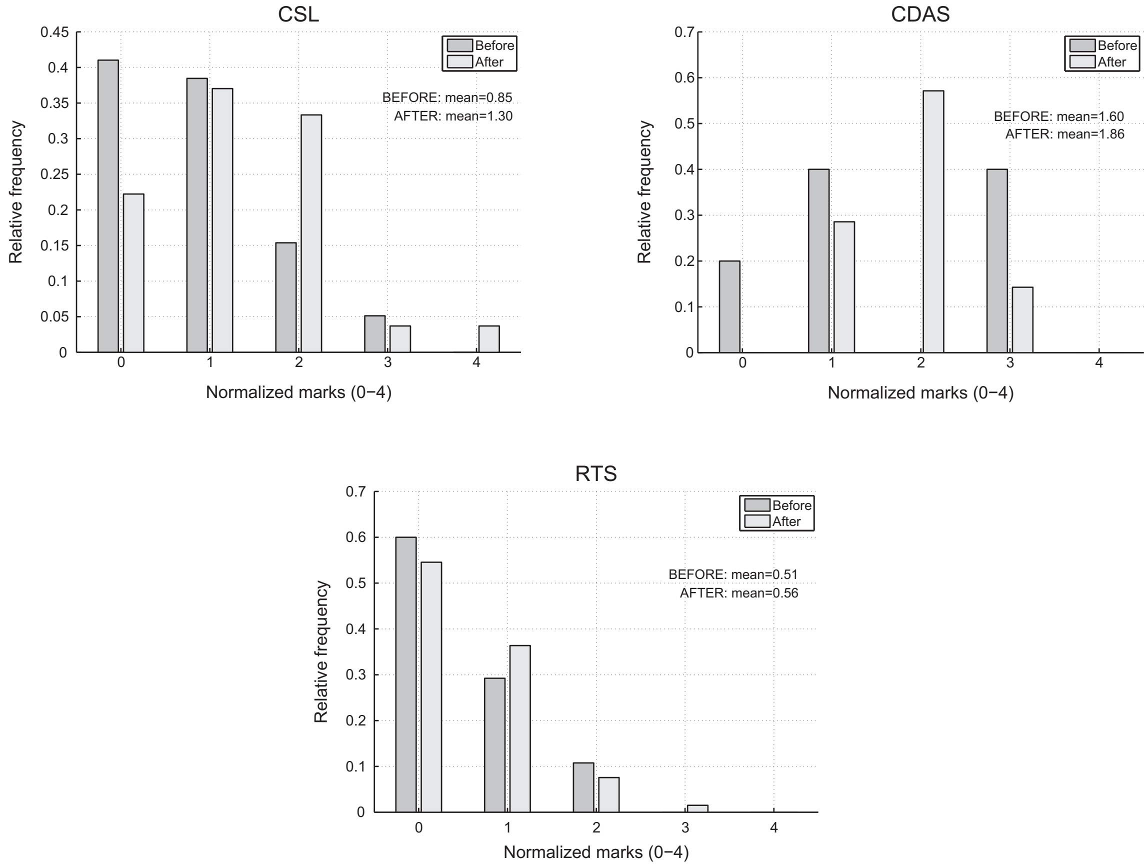


Fig. 5. Evolution of the scores obtained by the students in three of the subjects of this work. Marks are standardized: 0 corresponds to not passing the subject, 1–4 corresponds to passing with different levels (4 is rare in general, indicating top level of excellence). In the Spanish university system, there exist several times along an academic year when the students have exams; we have selected for the figure the same one in each year for both before and after: the moment closest to the end of the theoretical classes. The before/after standard deviations on the figures are: CDAS-1.34/0.69, RTS-0.69/0-71 and CSL-0.88/0.99.

To sum up, the LEGO Mindstorms NXT has seemed to be a good choice from a teaching perspective since it is attractive to students for regular classes as well as BSc/MSc thesis development. Its economic cost is well paid off, since its uses go further, and actually is being used also for a research task related to networked robots carried out in our Department.

In the following we present a more detailed statistical analysis of the subjects where more alumni have answered our questionnaires: Control Systems Laboratory and Real-Time Systems. Both include all the topics needed in the other subjects, thus they are highly representative of the results.

6.1.1. Control Systems Laboratory (CSL)

For this subject, we apply the practical exercises presented in Section 5.3, combined with a previous set of non-LEGO Matlab/Simulink based exercises.

In order to measure how the students felt using the robots in this subject, two anonymous questionnaires were made, one at the beginning of the semester, and another at the end. The first one had four closed questions centered on the previous knowledge of the students regarding some aspects (programming, robotics, etc.). The second repeated these four questions, so the students could measure its own advance in such issues; it also included eight additional questions related to the logistics part of the practical exercises (quality of the auxiliary material, and the like), as well as two free expression questions in order to know as much as possible about their real opinion of using LEGOs in the subject. Table 2 shows the questions included in the first questionnaire.

The statistical results are shown in Tables 3 and 4, with a difference on the number of students (36 before and 22 after) due to the fact that the Málaga University allows those students who are taking the subject for second (or further) time to have an extraordinary exam of the subject in December, just in the middle of the semester. Some students passed the exam, so they left the ordinary subject.

Table 2
Questions about the previous knowledge of CSL students.

1. - Rate from 1 (nothing) to 5 (much) your knowledge/competences on C programming.
2. - Rate from 1 (nothing) to 5 (much) your knowledge/competences on embedded systems (microcontrollers).
3. - Rate from 1 (nothing) to 5 (much) your knowledge/competences on controlling physical systems in order to obtain some desired behavior.
4. - Rate from 1 (nothing) to 5 (much) your knowledge/competences on any kind of robotic systems.

Table 3
Results of the before questionnaire of CSL students.

	Question 1	Question 2	Question 3	Question 4
Mean	3.22	2.69	2.28	1.64
Deviation	0.9	1.01	1	0.9
Skewness	−0.47	−0.04	0.66	2.05

The obtained data are positive. The mean after the semester has increased in all the questions; furthermore, the skewness is broadly near 0 or negative, which means that the answers are distributed symmetrically around the mean or at his right (that is, most of the answers correspond to high values).

It is remarkable that question No. 1 points out that the choice of the NXC language has been correct, since the students have previous knowledge about the C language and they can program the robot from the start without difficulties. Anyway, our goal for the next years will be to achieve a higher mean increase of every question in the after questionnaire.

Table 5 collects the remaining questions in the after survey, and their statistical results are gathered in Table 6.

It can be observed that most of the means of the questions are over 3 (the median of the levels offered as possible answers), and most of the skewness is negative, which means that most of the answers are values over the mean. If we analyze each question, we can conclude that, though the students do not consider that LEGO exercises are difficult (Question No. 7, mean 3.68), they do not complete these practical exercises as much as they fulfill the non-LEGO exercises (Questions No. 5 and No. 6, means 4.27 and 2.86, respectively). The students are pleased with the supporting material we have prepared, but we have detected that, though this material was available to them from the beginning of the semester, and despite we have warned them that they should get familiar to it as soon as possible in order to obtain a better understanding of the NXT platform, it was not utilized until the very moment they began to work with the robot, so the advance in the LEGO-based practical exercises was slower than expected. Since academic year 2010–2011, we have overcome the problem by providing them with the introductory practical exercise solved, along with the robot, from the beginning of the subject, so they can work and learn about the robot at the end of the each regular lab session, and are already trained when the LEGO-based practical exercises starts.

Finally, students are satisfied both with the use of the robot and the subject (Questions No. 9 and No. 10).

The last two questions of the survey are devoted to the free opinion of the students, as it is shown in Table 7.

Among all the submitted questions, we collect here a set of the most interesting ones:

- Do you think that the LEGO NXT robot is appropriate for this subject? Why?
  - Yes, because it allows the application of the concepts of the subject and the initiation to using sensors and actuators without facing design and/or assembling problems.
  - It is an agreeable way to apply the theoretical concepts we have acquired. Furthermore, it motivates me to keep on working in this field. On the whole, I think I have get the most out of the LEGO practical exercises than the other exercises.
  - Yes, since we can see the results of the previous practical exercises applied to the robot, and then check if it works as we expected.
  - Yes, because its use is easy and intuitive.
- Which aspects do you think would improve the subject? What did you miss? What would you change?
  - May be I would change the density of some practical exercises since some of them were very long. Some robot based practical exercises were slightly difficult, but in general I am very satisfied with the subject.
  - I would solve the practices solved in class by uploading the files to the Internet. In this way, we would speed up the progression of the classes. I would reduce the number of practical exercises. It would be interesting to get part of the final mark could be get with these exercises.
  - The time for developing some parts is not enough. The lab should be opened apart from the classes time.

6.1.2. *Real-Time Systems (RTS)*

In this subject the students have the chance to add some points to their final marks by preparing a voluntary work related to the LEGO NXT. In previous courses, due to the high number of pupils, it was impossible to offer enough works of this kind to cover the demand. The use of LEGO Mindstorms has made it possible: every single student interested in this voluntary work was assigned. The works consist of three session labs of one hour each, specifically focused on real-time systems. The students, grouped into pairs, performed the practical exercises, and finally answered a written questionnaire about their results in order to get their mark. In the web of the course they had different links available related to LEGO and NXC, as well as a specific datasheet for the laboratory exercises. The number of students taking part in these works was 119, which means a 300% increment if compared with the 26 students that could be assigned a voluntary work in the 2008–2009 course.

If we focus on how this has influenced the final grades of the students, we present here the obtained results in the February final exam call of 2009/2010 year. The number of students that showed up to this call was 66 (the low confidence of the students in their acquired skills is an issue to investigate): 30 of them passed the exam, and 36 failed it. We show the relationship between succeeding the exam and preparing the voluntary work in Table 8.

Table 4
Results of the after questionnaire of CSL students.

	Question 1	Question 2	Question 3	Question 4
Mean	3.59	2.77	2.32	1.95
Deviation	0.91	1.07	0.95	0.72
Skewness	0.12	−0.8	0.02	0.07

Table 5
Questions regarding to the progress of the subject in the after survey.

5. -	On average, until which point have you completed each **Matlab/Simulink** practical exercises? (1-I have not completed, on average, any, 5-I have completed, on average, all of them
6. -	On average, until which point have you completed each **LEGO** practical exercises? (1-I have not completed, on average, any, 5-I have completed, on average, all of them
7. -	Which degree of difficulty would you assign to the LEGO practical exercises? (1-very easy, 5-very complicated)
8. -	How do you rate the complementary material we have prepared for the practical exercises (guides, tutorials, wiki...)? (1-very bad, 5-very good)
9. -	Have you fulfilled your expectations about the use of the robots? (1-not fulfilled at all, 5-completely fulfilled)
10. -	Which is the degree of satisfaction with the subject this year? (1-not satisfied at all, 5-completely satisfied)

The probability of passing the course if the work has been prepared is 53.06%, whereas the probability of failing the course in that case is 46.94%, so it would seem that the voluntary work does not have a conclusive effect on obtaining a positive final grade. However, that is not a proper reasoning, since there are some other elements that have an influence on the fact of passing the subject, namely: intermediate exams passed, the fact that some students diminish their effort if such intermediate exams are positive, etc. The remarkable fact here is rather the probability of passing or failing the final exam *without* the voluntary work done: 23.53% and 76.47% respectively; that is, without the work, passing the exam is not very probable, while failing is more likely. We could deduce from this that the voluntary work serves as a detector of motivated students.

Finally, we pose an anonymous questionnaire to the alumni via the web of the subject, which was completed by 42 of the 121 students that attended the voluntary works. The questions we made, embracing different aspects of the subject and the works, are included in Table 9 in Appendix A due to its length, since their nature is very similar to the questionnaires previously presented for other subjects. Statistical data are presented in Table 10 in Appendix A too; two questions are not included in that table (#10, since it is a multiple choice one, and #21, since it is a free opinion one), and question #20 has less answers because not all the students have included the numerical response that we asked for.

We should pay attention to the most important questions related to the application of the robots in the subject:

- The purpose of questions #1 and #2 was to show how the students estimate their progress in the C language knowledge. According to the obtained results, they felt there is a considerable improvement.
- Questions #3 and #4 display the evolution of the knowledge of the students about real-time embedded systems, a core part of the RTS subject. In this case, the results are good, and the pupils have passed from a low average (1.9) and positive skewness to a better average (3.33) and negative skewness.
- Questions #6 and #7 evaluate the progress of the students related to the knowledge/competences about processing external signals of embedded systems in order to extract some information. The statistical data are good too: we advance from a 2.19 average and positive skewness to a 3.17 average and negative skewness. Again, the teaching possibilities of the robot are corroborated.
- Questions #15, #16 and #17 gather the opinion of the students about the theoretical and practical difficulty of the proposed works, and about their degree of completion. In both questions #15 and #16, the average is approximately 3, with near zero skewness, so we can conclude that the students agree that the difficulty is medium or slightly high. Question #17 gets a 3.74 average with negative skewness. So, it seems that the works are well calibrated in difficulty and duration.
- Question #18 analyses the quality and quantity of the knowledge that the students have acquired due to the voluntary works. The average is 3.33 with negative skewness, so it seems that the alumni appreciate the contents that have been presented in the works.
- Question #19 focus on the relationship between performing the voluntary works and the comprehension of the subject. The average is 2.9, which is not a bad result, but reveals that a bigger effort must be done by the teachers in order to strength the link between these works and the syllabus of the subject, so the former become more effective.
- Finally, question #20, though it has been answered properly by half the students, collects good results, since the students have fulfilled their expectations with a 3.5 average.

In the following we show the free opinions that we have found more interesting and representative of how the students feel:

- Which was your motivation for preparing the voluntary works?
  - My main reason was to get the extra grade, and the curiosity I felt about seeing embedded systems and particularly the LEGO Mindstorms.
  - To get the extra grade for the final exam. Performing the works has been a nice surprise, since I have seen with my eyes what a real-time system may be useful for.
  - Because they were interesting, since not many subjects offer works related to programming robots, and to get some extra grade.
  - I was mainly interested in seeing how an embedded system works and its utility, as well as the extra grade.
- Suggest anything you want about how the voluntary works have been organized.

Table 6
Results of the progress of the subject questions in the after survey.

	Question 5	Question 6	Question 7	Question 8	Question 9	Question 10
Mean	4.27	2.86	3.68	3.73	3.32	3.59
Deviation	0.77	0.89	0.84	0.94	0.95	1.1
Skewness	−0.53	−0.61	0.16	−0.93	−0.35	−0.74

Table 7
Free expression questions in the after survey.

11. -	Do you think that the LEGO NXT robot is appropriate for this subject? Why?
12. -	Which aspects do you think would improve the subject? What did you miss? What would you change?

Table 8
Relation between voluntary works and exam results.

	Final exam passed	Final exam failed
Voluntary work prepared	26	23
No voluntary work prepared	4	13

- ○ I think we were short of time for the works, o they appear to be short. To find a more clear relationship between the theory and the practice.
- ○ A little extra time in some of the works.
- ○ I would have liked to have one more work, with a difficulty between the second and the third work.
- ○ As my single suggestion, I would ask that the works could cover more days so we can deepen a little more.

6.2. Teaching results: the professor's point of view

The LEGO Mindstorms NXT was preliminary chosen as the hardware platform to develop our work mainly for two reasons: firstly, the advantages that this platform offers, such as price, ease of use, different programming options, and the variety of sensors that came with the basic kit; secondly, the knowledge we acquired through the work we developed with previous versions of the robot. In order to assess the final results of our teaching approach for the described experience, it is also necessary to analyze the NXT behavior under real job conditions, and after that we will be able to check if the initial prospects have been really fulfilled. In the following points we present our professor experience after two years of labor with the NXT:

- The maintenance tasks, which must assure that each robot is in good conditions before the beginning of any class, do not require neither user prior training, nor hardware resources. Most of these tasks, including the files deletion, charging of batteries, configuration changes and hardware tests, can be performed without connecting the NXT to a PC; a computer is only needed in order to perform some additional actions, like firmware restoring. This is an important time saving feature.
- The LEGO Mindstorms NXT software is robust: in two years, the operating system has rarely failed, and seldom been corrupted. When a problem occurs, it can easily be solved.
- Regarding the hardware of the robot, we noticed that the engines are not very tough, and we could detect some noise problems due to exhausted mechanics during long-time observations, for example, when testing a PID line follower.
- The mechanical structure (wheels, axles) of the physical configuration we have selected is not very strong either, so the students may obtain different results in a lab session depending on which robot they work with. This issue can be useful for the students to experience and understand one of the main difficulties that every robotic system must deal with: the real world is not perfect.
- As we could see in the aforementioned surveys, students have been able to adapt easily themselves to the robot, as well as to the programming environment.
- The sensors included in the basic kit of the NXT do not allow to perform advanced tasks, so if we want to use the robot in more complex tasks, such as a BSc/MSc thesis or in research projects, it may be necessary to acquire new sensors or build them ourselves.
- Before starting every session, it is essential to check that the batteries of the robot are completely charged, especially if we are going to use them for 2 or 3 h. Working like that, we will avoid possible malfunctions of the sensors or the actuators. This means that the charging task must be properly organized every week, which can be really complicated if the NXT sessions are too close in the classes schedule. In order to avoid teachers overload, this job should be carried out by laboratory assistants. Another possibility, though more expensive, is to have enough spare batteries ready and previously charged, and use them to replace the batteries of the Mindstorms when they are empty.
- As above-mentioned, it is advisable to remove all the program files stored into the NXT by the students, which is a simple but tedious task, and would mean overworking the teacher if no laboratory assistant is available.

To sum all these points up, we can conclude that using the LEGO NXT to our purposes has more benefits than drawbacks, and it fits well to the development of the practical exercises we have designed.

7. Conclusions and future work

We have presented in this paper a teaching innovation project devoted to the use of LEGO Mindstorms NXT robots in subjects related to data acquisition, control systems and real-time systems in several undergraduate engineering courses at Málaga University. Along with a description of the different laboratory exercises we have prepared for these subjects, we have also analyzed some surveys our pupils have answered, in order to record how they feel about using this educative robotic tools for their training.

Our experience using these robots has been really positive. Our students find the lab sessions more attractive, and they solve the proposed practical exercises more enthusiastically; this perception is statistically supported by the results of the surveys we have conducted in different subjects and, more slightly, by the scores obtained by the students, which we believe comes from a higher motivation of the platform compared with traditional approaches. On our behalf, we have found that the NXT is a nearly optimal platform for engineering subjects in terms of cost, robustness and versatility. Thus, the proposal achieves simultaneously all the requirements explained in Section 1.

Though our results are related to the 2008–2009 and 2009–2010 courses, we have continued our work with these robots since then, not only in some of the undergraduate level subjects previously commented, but in new post-graduate level courses: for example, LEGO Mindstorms NXT is now also used for teaching mobile robotics aspects in some subjects of the Mechatronics Engineering Master (Master Universitario en Ingenieria Mecatronica, 2011) offered by Systems Engineering and Automation Department (2011) at Málaga University. Apart from extending the application area of the robots to new subjects, we also plan to improve the results we have obtained until now according to the data provided by the surveys, as well as to include new educative techniques, like contests or YouTube devoted channels, to our classes.

Acknowledgments

This work has been developed under the teaching project “PIE08-008 Innovación en las asignaturas relativas a Sistemas de Control mediante robots LEGO MINDSTORMS NXT” supported by the “Proyectos de Innovación Educativa 2008-2010 - Elaboración e implantación de nuevos planes de estudio” program at Málaga University. The authors are also grateful to the E.T.S.I. Informática and to the Systems Engineering and Automation Department, both at Málaga University, for their support and financial help.

Appendix A

The questionnaire presented to the students of the real-time course is shown in Table 9. Statistical data are presented in Table 10. Two questions are not included in the latter: #10, since it is a multiple choice one, and #21, since it is a free opinion one; question #20 has less answers because not all the students have included the numerical response that we asked for.

Table 9
Questions asked to the RTS students.

1. - Rate from 1 to 5 your knowledge about C programming language at the beginning of the course.
2. - Rate from 1 to 5 your knowledge about C programming language after preparing the voluntary NXC works (1-I have not improved at all, 5-I have learned very much).
3. - Rate from 1 to 5 your knowledge about embedded systems before the voluntary works.
4. - Rate from 1 to 5 your knowledge about embedded systems after the voluntary works.
5. - Rate from 1 to 5 your knowledge/competences about about computer/microcontroller control of systems with temporal requirements after the voluntary works.
6. - Rate from 1 to 5 your knowledge/competences, before the voluntary works, about processing signals acquired from the outside of a embedded system, in order to analyze them and extract some information.
7. - Rate from 1 to 5 your knowledge/competences, after the voluntary works, about processing signals acquired from the outside of a embedded system, in order to analyze them and extract some information.
8. - Rate from 1 to 5 your knowledge/competences about any kind of robotic systems.
9. - In our current society, which role (1-none, 5-very wide) do you think embedded systems play?
10. - For what do you think that a computer can be used inside of, or associated to, a physical real system that has a significant presence in our current society (you can write down more than one answer)?
11. - Which theoretical complexity (1-none, 5-very much) do you think it has the study of including a computer into a physical system?
12. - Which practical complexity (1-none, 5-very much) do you think it has the inclusion of a computer into a physical system?
13. - How different (1-no difference, 5-totally different) do you think is the software of a computational embedded system compared to the software of a computational desktop or sever system?
14. - How would you rate your competences (1-none, 5-very much) about development environments for embedded systems (host/target) after the voluntary works?
15. - Rate the theoretical difficulty (1-none, 5-very much) you have found in the voluntary works?
16. - Rate the practical difficulty (1:none, 5:very much) you have found in the voluntary works?
17. - On average, until which point have you completed the voluntary works in the planned sessions? (1-I have not completed any, 5-I have completed all of them.
18. - Rate from 1 to 5 the quantity and quality of the knowledge you have acquired preparing the voluntary works.
19. - How much has helped you preparing these works for studying/understanding the subject? (1-nothing, 5-very much).
20. - Which was your motivation for preparing the voluntary works? How much (1-nothing, 5-completely) have been your expectations fulfilled?
21. - Suggest anything you want about how the voluntary works have been organized.

Table 10
Results of the questionnaire of RTS students.

Question	1	2	3	4	5	6	7	8	9	11	12	13	14	15	16	17	18	19	20
Total	42	42	42	42	42	42	42	42	42	42	42	42	42	42	42	42	42	42	28
Average	3.17	2.74	1.9	3.33	3.07	2.19	3.17	3.36	4.48	3.71	4.12	3.57	3.07	3	3.05	3.74	3.33	2.9	3.5
Deviation	0.91	1.04	0.98	1.12	1.02	1.15	1.12	0.93	0.63	0.83	0.8	0.99	0.92	0.88	1.08	1.06	0.93	1.1	1.1
Skewness	−0.35	0.15	1.17	−0.6	−0.44	0.41	−0.56	−0.03	−0.81	0.06	−0.52	0.03	0.05	0	0.15	−0.34	−0.16	−0.03	−0.4

References

Arduino. (2011). http://www.arduino.cc/.

Aslam, D. M., Cao, Z., & Rostamzadeh, C. (2008). Innovative engineering education using programmable LEGO robotic VD graaf generators. *Proceedings of the ASEE Zone 1 Conference*.

Behrens, A., Atorf, L., Schwann, R., Neumann, B., Schnitzler, R., Ballé, J., et al. (2010). MATLAB meets LEGO Mindstorms – a freshman introduction course into practical engineering. *IEEE Transactions on Education, Vol. 53*(2), 306–317.

Benedetelli, D., Casini, M., Garulli, A., Giannitrapani, A., & Vicino, A. (2009). A LEGO Mindstorms experimental setup for multiagent systems. In *18th IEEE international conference on control applications, part of the multi-conference on systems and control* (pp. 1230–1235).

Bricx Command Center 3.3. (2011). http://bricxcc.sourceforge.net/.

Cliburn, D. C. (2006). Experiences with the LEGO Mindstorms throughout the undergraduate computer science curriculum. In *36th ASEE/IEEE frontiers in education conference, San Diego (CA)*.

Cuéllar, M. P., & Pegalajar, M. C. (2011). *Design and implementation of intelligent systems with LEGO Mindstorms for undergraduate Computer Engineering*. Published online in Wiley Online Library, doi:10.1002/cae.20541.

CV-UMA. (2010). http://informatica.cv.uma.es/course/view.php?id=1099.

Fagin, B., & Merkle, L. (2003). Measuring the effectiveness of robots in teaching computer science. In *Proceedings of the 34th SIGCSE technical symposium on computer science education* (pp. 307–311).

Feedback plc. (2011). *Educational equipment, time and attendance and access control products from feedback*. http://www.feedback-group.com/.

Fernández Panadero, C., Villena Román, J., & Delgado Kloos, C. (2010). Impact of learning experiences using LEGO Mindstorms in engineering courses. In *IEEE conference on education engineering (EDUCON), 14–16 April* (pp. 503–512).

Fiorini, P. (2005). LEGO kits in the lab. *IEEE Robotics & Automation Magazine, Vol. 2*(4), 5.

Gage, A., & Murphy, R. (2003). Principles and experiences in using LEGO's to teach behavioral robotics. In. *33rd ASEE/IEEE frontiers in education conference, Boulder, CO, Vol. 2*.

Galvan, S., Botturi, D., Castellani, A., & Fiorini, P. (2006). Innovative robotics teaching using Lego sets. In *Proceedings of the 2006 international conference on robotics and automation, Orlando USA* (pp. 721–726).

Gawthrop, P. J., & McGookin, E. (2009). Using LEGO in control education. In *7th IFAC symposium on advances in control education, Madrid, Spain*.

Gómez-de-Gabriel, J. M., Mandow, A., Fernández-Lozano, J., & García-Cerezo, A. (2011). Using LEGO NXT mobile robots with LabVIEW for undergraduate courses in mechatronics. *IEEE Transactions on Education, Vol. 54*(1), 41–47.

Greenwald, L., Artz, D., Mehta, Y., & Shirmohammadi, B. (2006). Using educational robotics to motivate complete AI solutions. *AI Magazine, Vol. 27*(1), 83–95.

Grega, W., & Pilat, A. (2008). Real-time control teaching using LEGO MINDSTORMS NXT ROBOT. *Proceedings of the International Multiconference on Computer Science and Information Technology*625–628.

Hamada, M., & Sato, S. (2011). A game-based learning system for theory of computation using Lego NXT robot. *Procedia Computer Science, Vol. 4*, 1944–1952.

Han Kim, S., & Wook Jeon, J. (2009). Introduction for freshmen to embedded systems using Lego Mindstorms. *IEEE Transactions on Education, Vol. 52*(1), 99–108.

Hansen, J. C. (2007). *NXT power programming*. Variant Press.

Harbour, D., & Hummel, P. (2010). Migration of a robotics platform from a freshman introduction to engineering course sequence to a sophomore circuits course. In *40th ASEE/ IEEE frontiers on education conference*, pp. F4C-1–F4C-6.

Jaksic, N., & Spencer, D. (2007). An introduction to mechatronics experiment: LEGO Mindstorms next urban challenge. *American Society for Engineering Education*, Available at www.icee.usm.edu/icee/conferences.

Karp, T., Gale, R., Lowe, L. A., Medina, V., & Beutlich, E. (2010). Generation NXT: building young engineers with LEGO. *IEEE Transactions on Education, Vol. 53*(1), 80–87.

Khamis, A. M., Rodríguez, F. J., & Salichs, M. A. (2003). Remote interaction with mobile robots. *Autonomous Robots, Vol. 15*(n. 3), 267–281.

Kim, Y. (2011). Control systems lab using a LEGO Mindstorms NXT motor system. *IEEE Transactions on Education, Vol. 54*(3), 452–461.

Klassner, F., & Anderson, S. D. (June 2002). LEGO Mindstorms: not just for k-12 anymore. *Robotics and Automation Magazine*12–18.

Klassner, F., & McNally, M. (2007). Demonstrating the capabilities of Mindstorms NXT for the AI curriculum. In *Proceedings of AAAI spring symposium (robots and robot venues): Resources for AI education track*, AAAI Technical Report SS-07–09(2007).

Lau, K. W., Tan, H. K., Erwin, B. T., & Petrovic, P. (1999). Creative learning in school with LEGO programmable robotics products. In. *29th ASEE/IEEE frontiers in education conference, Vol. 2*, pp. 12D4/26–12D4/31.

LEGO Engineering. (2011). http://legoengineering.com/.

NXTPIE. LEGO NXT AND CONTROL SYSTEMS TEACHING. (2010). http://nxtpie.wikispaces.com.

LEGO Mindstorms: Home. (2011). http://mindstorms.lego.com.

Lew, M. W., Horton, T. B., & Sherrif, M. S. (2010). Using LEGO MINDSTORMS NXT and LEJOS in an advanced software engineering course. In *23rd IEEE conference on software engineering education and training* (pp. 121–128).

Lucas-Nülle. (2011). *Lucas-Nülle training systems for vocational training and didactic*. http://www.lucas-nuelle.com/.

Lund, H. H., & Pagliarini, L. (2000). RoboCup Jr. with LEGO Mindstorms. In. *Proceedings of the 2000 IEEE international conference on robotics and automation, San Francisco, Vol. 1* (pp. 813–819).

Maher, R. C., Becker, J., Sharpe, T., Peterson, J., & Towle, B. A. (2005). Development and implementation of a robot-based freshman engineering course. *Proceedings of the American Society for Engineering Education Annual Conference & Exposition*.

Máster Universitario en Ingeniería Mecatrónica. (2011). http://www.isa.uma.es/C6/mastermecatronica/default.aspx.

Mathworks. (2011a). *Programming the LEGO MINDSTORMS NXT robotic toolkit with Matlab & Simulink*. http://www.mathworks.com/products/demos/rtw/lego/.

Mathworks. (2011b). http://www.mathworks.com.

Matlab Central. (2011). *Embedded coder robot NXT demo*. http://www.mathworks.com/matlabcentral/fileexchange/13399.

Miller, D. P., Nourbakshs, I. R., & Siegwart, R. (2008). Robots for education. In Siciliano., & Khatib. (Eds.), *Handbook of robotics*. Springer.

Mota, M. I. G. (2007). Work in progress – using Lego Mindstorms and Robolab as a mean to lowering dropout and failure rates in programming course. In *37th ASEE/IEEE frontiers in education conference* (pp. F4A1).

Nagchaudhuri, A., Singh, G., Kaur, M., & George, S. (2002). LEGO robotics products boost student creativity in pre-college programs at UMES. In. *32nd annual frontiers in education, FIE 2002, Vol. 2*, pp. S4D1–S4D6.

NBC/NXC – NeXT Byte Codes and Not eXactly C. (2011). http://bricxcc.sourceforge.net/nbc/.

NI LABVIEW. (2011). http://www.ni.com/labview/.

Oliver, J., Toledo, R., & Valderrama, E. (2010). A learning approach based on robotics in computer science and computer engineering. In *IEEE conference on education engineering (EDUCON), 14–16 April* (pp. 1343–1347).

Pomalaza-Ráez, C., & Henry Groff, B. (2003). Retention 101: where robots go, students follow. *Journal of Engineering Education, Vol. 92*(n. 1), 85–90.

Ranganathan, P., Schultz, R., & Mardani, M. (2008). Use of Lego NXT Mindstorm brick in engineering education. *Proceedings of the ASEE North Midwest Sectional Conference*.

Sharad, S. (2007). Introducing embedded design concepts to freshmen and sophomore engineering students with LEGO Mindstorms NXT. *IEEE International Conference on Microelectronics Systems Education*119.

Sluka, J. (2009). *A PID controller for LEGO Mindstorms robot*. http://www.inpharmix.com/jps/PID_Controller_For_Lego_Mindstorms_Robots.html.

Summet, J., Kumar, D., O'Hara, K., Walker, D., Ni, L., Blank, D., et al. (March 2009). Personalizing CS1 with robots. *ACM SIGCSE Bulletin, Vol. 41*(1).

Systems Engineering and Automation Department. (2011). http://www.isa.uma.es.

Touretzky, D. S., & Tira-Thompson, E. J. (2010). The Tekkotsu 'Crew': teaching robot programming at a higher level. In *Proceedings of the 24th AAAI conference on artificial intelligence* (pp. 1908–1913).

Valera, A., Vallés, M., Fernández, A., & Albertos, P. (2009). Platform for the development of mechatronical practical works based on LEGO Mindstorms NXT robot. In *18th international conference on control applications* (pp. 1224–1229).

Wang, E., LaCombe, J., & Rogers, C. (2004). Using LEGO bricks to conduct engineering experiments. *Proceedings of the American Society for Engineering Education Annual Conference & Exposition*, session 2756.

Williams, A. B. (2003). The qualitative impact of using LEGO MINDSTORMS robots to teach computer engineering. *IEEE Transactions on Education, Vol. 46*(n. 1), 206.