Using LEGO Robots with LabVIEW for a Summer School on Mechatronics

Alfonso García-Cerezo, Jesús Gómez-de-Gabriel, Jesús Fernández-Lozano, Anthony Mandow,
Victor F. Muñoz, Fernando Vidal-Verdú*, and Klaus Janschek**

Dpto. Ingeniería de Sistemas y Automática, Universidad de Málaga, 29071 Málaga, Spain
* Dpto. Electrónica, Universidad de Málaga, 29071 Málaga, Spain
** Inst. f. Automatisierungstechnik, Technische Universität Dresden, Dresden, Germany
Email: gcerezo@ctima.uma.es, Tel: (+34) 952 132087, Fax: (+34) 952 131413

Abstract— **This paper reports on the experience of the 2008 International Summer School on Mechatronics, jointly organized by the University of Málaga (Spain) and the Technical University of Dresden (Germany). An important part of the hands-on practice and two student competitions have been based on the LEGO Mindstorms NXT Set. To stimulate lab work with representative general purpose software tools, LabVIEW and the NXT add-ons have been used. The paper proposes basic LabVIEW structures for several LEGO case study practices. A description of the course as well as an assessment on student competences are also included.**

I. Introduction

Mechatronics is bound to play a crucial role in engineering education as a unifying, interdisciplinary, and intelligent science paradigm. As pointed out in [1], mechatronics ”focuses on the process of learning linked with actions rather than teaching, and the learning process is directly coupled with thinking.” In this sense, project-based learning emerges as a useful tool to develop complementary incorporation of vertical (i.e., abstract and procedural) and lateral (i.e., search and brainstorming) thinking [2]. This model is in tune with the challenges of the European Higher Education Area [3].

Mechatronic platforms such as LEGO robot kits are increasingly being used with educational purposes, ranging from secondary school, where the accent is on basic technology and problem solving [4], to more specialized university courses [5] [6].

Programming tools offered by LEGO (i.e, RoboLAB and NXT Education Software) have limited flexibility, but they are intuitive and visual so as to be used by students of any technological background. Nevertheless, some higher education courses have adopted more general and flexible solutions, like Java, by changing the original firmware [7].

Moreover, LEGO NXT units directly support National Instrument's Laboratory Virtual Instrument Engineering Workbench (LabVIEW). In fact, this is the underlying software structure of the specific LEGO programming tools. LabVIEW advantages include its wide use in the professional control community, extensive support for accessing instrumentation hardware, simplified definition of user interfaces, and structured programming. For these reasons, this software tool is found in university courses, such as electronics [8] and control [9], as well as in remote laboratories [10] [11]. Moreover, this development environment is employed in actual mechatronics/robotics applications (e.g., [12]).

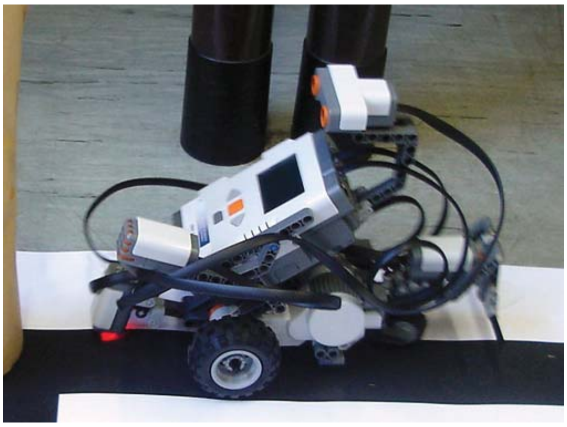


Fig. 1. LEGO NXT mobile robot.

In this paper we report on the experience of the 2008 Summer School on Mechatronics, jointly organized by the Technical University of Dresden (Germany) and the University of Málaga (Spain). Learning objectives of this two-week course for advanced engineering undergraduates included mobile robotics, control architectures, sensors and actuators, fuzzy logic, and project development. The paper describes lab work and student competitions based on LEGO NXT Mindstorm kits (see Fig. 1) and LabVIEW, a combination that fosters design and experimentation with real hardware and representative software. Basic LabVIEW structures for typical mobile robot problems are proposed in the paper so that they can be provided as a model to students with a non-proficient background in LabVIEW.

The paper is organized as follows. After this introduction, an overview of the Dresden-Málaga Joint Summer School is presented in Section II. A brief description of the LEGO NXT unit and specific LabVIEW programming tools is given in Section III. Then, basic LabVIEW solutions are proposed in Section IV for case lab projects on tele-operation, behavior-based navigation, and fuzzy control. Section V offers an assessment of educational results. Finally, the last sections and are devoted to conclusions, acknowledgements and references.

II. Summer School Overview

The Joint Summer School on Mechatronics started in 2007 as a first collaboration between two European research groups on Mechatronics: one from the University of Málaga and the other from the Technical University of Dresden. The idea of organizing a joint educational effort emerged given the similarities and complementarities found in the research interests and teaching curricula of both groups.

Both universities are engaged in different stages of the implementation of the European Higher Education Area (EHEA), an effort to create more comparable, compatible and coherent systems of higher education in Europe. Therefore, it was decided that the outline of the course should follow the new education model proposed for the EHEA, which is based on competences [3]. Competences are understood in this scenario as a dynamic combination of knowledge, understanding, skills and abilities. In this model, the final object of an educational programme should be fostering competences.

Competences can be distinguished between subject-specific and generic. The scope of the Summer School, with a limited duration and addressed to engineering students in their advanced years, has been focused on subject-specific ones. Particularly, the main goal was to introduce the students to mobile robotics and, at the same time, to extend their standard curricula by using this subject as a platform for other specific engineering areas. Thus, the courses comprise from Mobile Robot Architectures and Mobile Robot Control to Intelligent Robot Control and Hardware Architectures for Mobile Robotics, but also key technologies and techniques not always found in the Mobile Robotics curricula, like Internet based Tele-operation, 2D-Correlation Processing, VLSI-Design or Algorithm Mapping.

The Summer School was organized around ten teams of two mixed students (i.e., one from each university), with a duration of two weeks, the first one in Dresden and the second in Málaga. Each week consisted of four eight-hour courses (four for theory lectures, and four for lab work). The fifth day was devoted to a competition related to some of the week's topics. The rules for the competition were stated at the beginning of the week. The official language for the Summer School was English.

In order for the students to take advantage of such an intensive program, practical work should be arranged in a way that competences of every course are strengthened by lab work on the following days, whenever this is possible. This way, the courses can be planned incrementally: for instance, basics of sensor use are taught of the very first course, while in a second one the focus can be on controlling robots (using sensor information that they already know how to extract). Furthermore, to make the students more concerned about the evolution of their competences, robotic competitions play an important role in the Summer School.

Finding a common platform, available for practical work in different fields was a problem with many restrictions that was solved by using the LEGO NXT platform plus LabVIEW. To this end, each two-student team was assigned a LEGO Mindstorms Education Base Set (#9797) at the beginning of the Summer School. Whenever it was possible, the practical work of the courses was prepared around this combination, in such a way that it could drive the students preparing their projects for the competitions. In other courses, problems related to mobile robotics were presented as examples for motivation, even though different lab-work solutions were adopted to avoid forcing unrealistic practices or too large workloads. For these courses, oral or written examinations were arranged to assess fulfilment of the desired competences.

TABLE I

Course contents related to LEGO/LabVIEW

Subject	**Practice**	**Program Model**	**Lab Hours**
Prerequisite	LabVIEW introduction Tutorial	Generic	6
Mobile Robot Architectures	Introduction to LabVIEW/Lego NXT programming: Sensors reading and basic program structure.	NXT Toolkit	2
	Mobile robot control basics: Vehicle kinematics and state machines.	NXT Direct	2
Mobile Robot Control	Teleoperated vehicle control: User interface and advanced programming structures.	NXT Direct	2
	Autonomous mobile robot control: Line following.	NXT Toolkit	2
Competition	Line following and obstacle detection	NXT Toolkit	4
Intelligent robot control	Special sensors	NXT Direct	2
	Fuzzy logic control	NXT Toolkit	2
Competition	Wall following, goal seeking, and obstacle avoidance	NXT Toolkit	4

The courses that took part of the LEGO/LabVIEW experience in the 2008 Summer School are described in Table I. As a pre-requisite, students had to study a LabVIEW introduction tutorial so that they had a basic knowledge of this tool prior to the courses. Section IV provides more details about the LEGO/LabVIEW lab-work.

III. LabVIEW and LEGO NXT

This section describes the LEGO NXT hardware as well as the LabVIEW software platforms used for the lab work in the Mechatronics Summer School.

Regarding hardware, the LEGO NXT main unit is a battery powered module based on an 32-bit ARM micro-controller with an LCD display, 4 input ports for sensors, and 3 input/output ports for servo-motors. The sensor set includes two touch switches, an analog light sensor, a microphone, a range sonar, and three servo-motors. A variety of mechanical components and pieces allow for different physical arrangements and kinematic configurations. The NXT module allows communication with a PC either by Bluetooth or by USB cable. This can be used to access NXT services such as program downloading and monitoring.

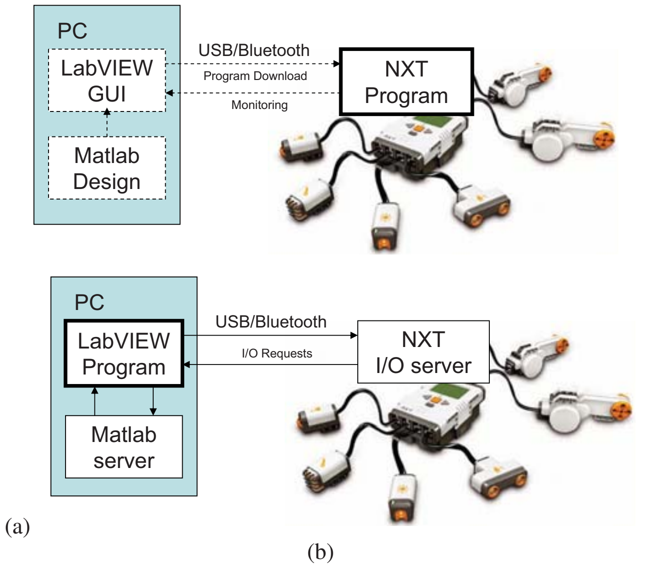


Fig. 2. NXT programming modes with LabVIEW. (a) NXT Toolkit. (b) NXT Direct.

The programming environment consists of the LabVIEW 8.2.1 version plus the LabVIEW Lego NXT add-on. Furthermore, some programming features of Matlab have also been used. The NXT add-on includes two different NXT-specific VI libraries that support two alternative programming models (see fig. 2):

- *Stand-alone NXT Toolkit*. The project target is the NXT main unit, so the compiled programs are downloaded to the robot. Then, program execution can be either off-line (i.e., stand-alone) or supervised from the PC for debugging purposes. In the latter case, a standard LabVIEW Front Panel GUI can be defined with both controls, to change program signals, and indicators. This programming model supports a subset of the standard labVIEW function set with additional constraints, such as the lack of floating point support and limited memory. Special VIs are found for the input/output devices of the NXT units. Other PC applications, such as Matlab, can be used off-line for controller design purposes only. Note that unlike specific NXT languages, this is standard LabVIEW, so experience from other applications can be used here, and viceversa.
- *NXT Direct Commands* where the PC is the execution target, so the robot cannot operate independently. The full LabVIEW resources and the PC computer power can be used. The NXT Direct Commands Library incorporates VIs to access NXT services such as sensor reading, motor control, file handling, and device management. This server is part of the standard robot firmware. This model is very interesting for applications that require additional resources like Matlab, ActiveX components, special I/O devices like joysticks, or extra computational power and memory.

IV. Lab Work Case Studies

This section proposes basic LabVIEW solutions for the lab-work contents of some of the courses enumerated in Table I.

A. Basic control and Tele-operation.

The introductory courses are devoted to the basics of mobile robot control. It also presents simple programming of sensor readings and motor control with the NXT Toolkit. This first robotic hands-on experience consists on assembling a basic differential drive vehicle configuration. Then, students develop a simple controller for straight motion, obstacle detection with the touch sensor, and turning. With this purpose, a kinematic model of the vehicle has to be programmed.

On the other hand, lab work for the Tele-operation course also serves as an introduction to the NXT Direct library programming model. The goal of this practice is to use the wireless bluetooth connection to safely tele-operate a mobile platform. A proposed solution to a basic exercise is presented in Fig. 3, which also illustrates a state-based control architecture.

Specific NXT VIs can be seen along the horizontal signal flow at the top of Fig. 3. This implies a sequential execution composed of a device search, initialization, sensor reading, actuator outputs, and device release. These blocks provide a software interface for the robot communication system.

The rest of the diagram illustrates the general structure of a control loop, which consists on a main loop with a timer. At its left, there are blocks for initialization of devices, and also for state variables (i.e., linear and angular speeds, and the controller state) that are remembered between iterations by means of shift registers. Inside the loop, a case structure is used for programming the different controller states. Besides, a common part consists of the kinematic model and the motor commands.

The controller state shown in the figure corresponds to tele-operation, which uses the keyboard for modifying the robot's angular and linear speeds. Note that a collision detection based on the sonar sensor has also been incorporated into the tele-operation mode.

B. Behavior-based navigation

Starting from the same state-based control architecture approach presented above, this practice pursues implementation of a line following behavior. In this case the stand-alone NXT Toolkit programming model is used (See Fig. 4). The line following behavior is based on a single analog light sensor so the control method is designed to follow the edge (left or right) of a thick line. Specific NXT VIs are used to read the light and ultrasonic sensors, synchronize differential drive outputs, and set loop timing.

The basic proportional control model is supplied to the students. Thus, they can concentrate on the hardware setup, controller parameters adjustment, and improvement of this control system. The solution proposed in Fig. 4 includes a

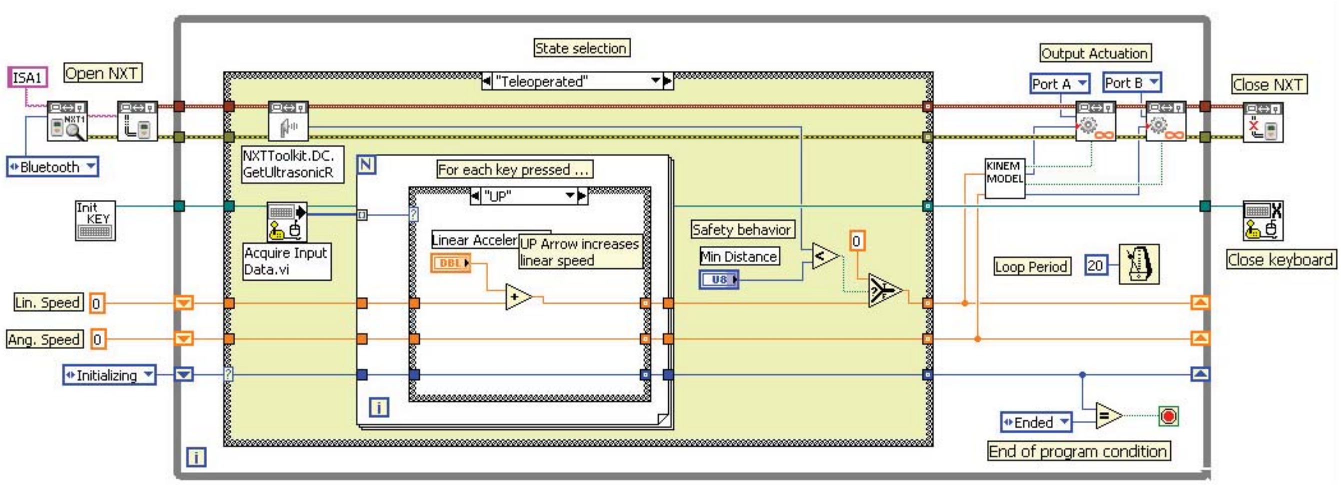


Fig. 3. Labview diagram for tele-operation example.

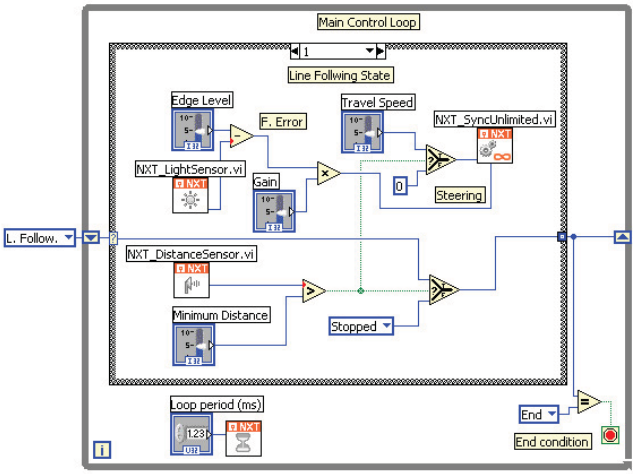


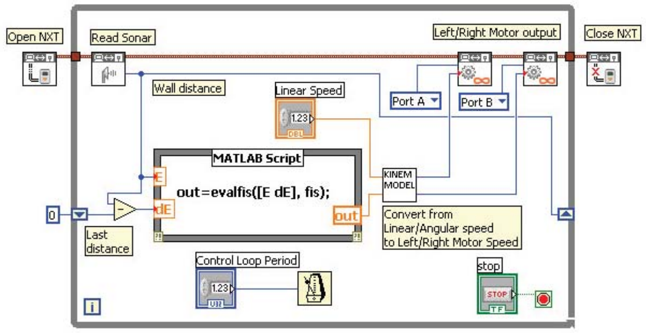
Fig. 4. Labview diagram for line following with the light sensor.

final state condition based on the sonar sensor. This state-based approach allows for flexible behavior sequences, which are useful for the student competitions.

C. Fuzzy Logic

The course on fuzzy logic control makes extensive use of the widely known Matlab Fuzzy Inference System toolbox. For the lab work, the two NXT LabVIEW programming modes allow to explore different ways to implement the resulting fuzzy systems.

Firstly, with the NXT direct library, the Matlab fuzzy toolbox can be applied to the Lego NXT without computing constraints. As can be seen in fig. 5, the Matlab Script structure is used inside the program main loop, in order to explicitly evaluate the fuzzy function. This Structure makes a call to the Matlab server, so a version of this software must be properly installed in each student PC. The first time this program executes, it takes a long time because of the initial load of the main Matlab program, but subsequent invocations are faster (less than 4 milliseconds on a laptop for a three inputs fuzzy system), so it can be used inside the main control loop. The rest of the control loop is composed by the sensor reading and input variable calculations, and the inverse kinematic model for the specific vehicle.



out=evalfis([E dE], fis);

Fig. 5. Labview diagram for fuzzy control with NXT Direct.

Secondly, autonomous execution is possible with the NXT Toolkit. In this sense, wall following control using fuzzy logic is presented in Fig. 6. The Matlab fuzzy toolbox has been used to obtain a static (fixed point) inference matrix. By means of a copy/paste method, the matrix is converted into a static LabVIEW array, so it can be used by the NXT stand-alone microcontroller. The real-time process has to calculate the fuzzy inputs upon the sensor readings and then the right array indexes should be obtained taking into account the input variable ranges and the chosen matrix resolution. The resulting element(s) is then the output(s) from the fuzzy controller that may need to be converted or scaled in order to be sent to the actuators. In this case the fixed point angular speed has to be converted to the left and right wheel speeds.

D. Competitions

The purpose of the competitions, which were not meant to be neither complex nor tight, was to maintain a high

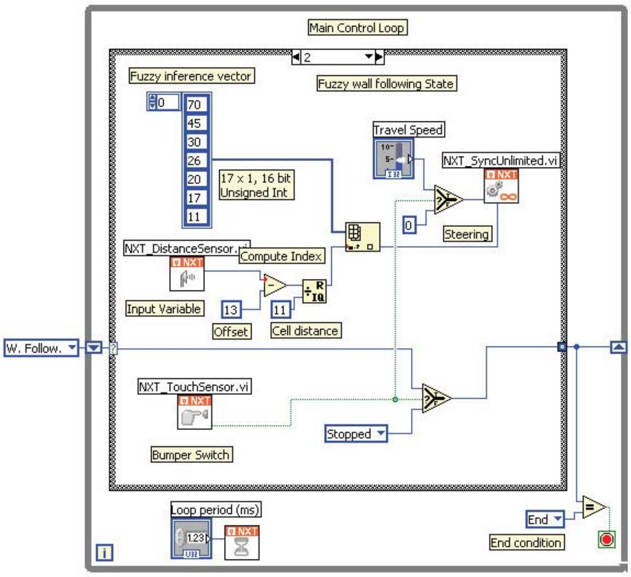
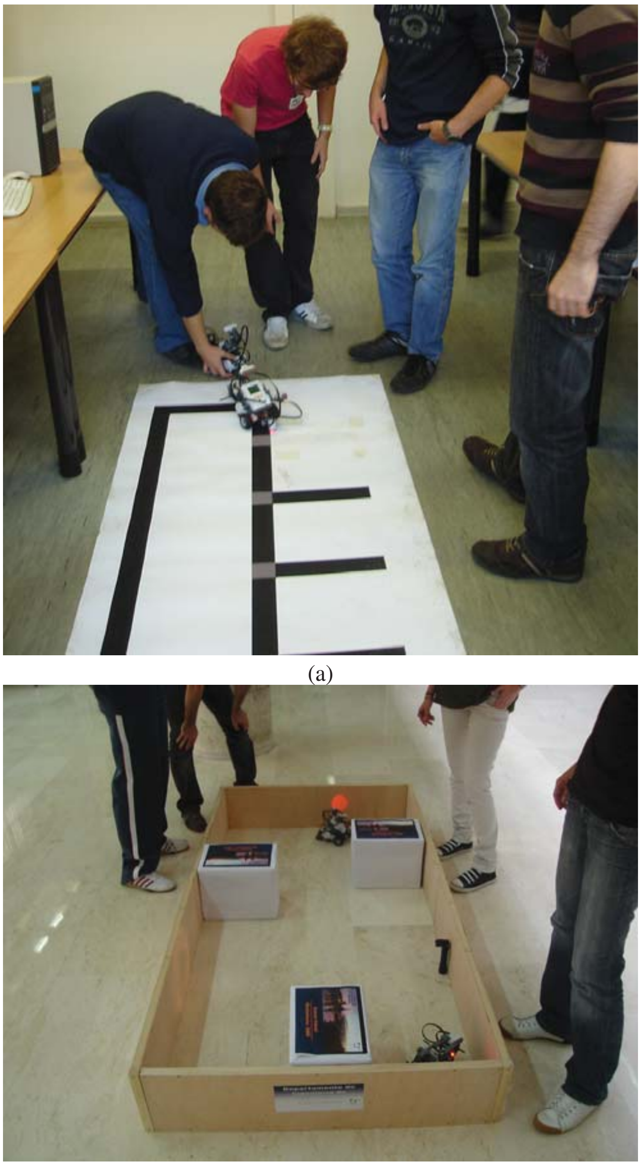


Fig. 6. Labview diagram for sonar-based fuzzy wall following.



(a)

(b)

Fig. 7. Two competition layouts. (a) Line Following. (b) Goal seeking.

student motivation during the courses. To this end, the main guidelines and rules were given at the beginning of each week so that the students could ponder the applicability of both lectures and practical work.

In the first competition (see Fig. 7a), which relates to practical work described in subsections IV-A and IV-B, the goal was following a known path defined by a thick black line on the ground, including 90° turnings. The path had to be completed in both directions, which implied turning back when a landmark obstacle was found at the end of the line. Scores were obtained for successfully passing over path landmarks from the best two out of three opportunities. We found that most students had been working on the competition goals after-hours during the week, so they used the competition lab time mainly for fine-tuning and improvement. All teams achieved the full score by the third run.

The second competition was designed to pose a more difficult challenge. The task was now defined as wall following while avoiding obstacles in unknown positions (see Fig. 7b), where the students could incorporate the practical work described in subsection IV-C. The task specification included general environment features like dimension and obstacle size. A light source identified the goal point, which had to be reached and acknowledged by some special action (e.g., a beep). Eventually, the time points decided the winner, since all teams successfully reached the goal.

V. Results and Student Feedback

Data collection was made through a survey. After finishing the Summer School, the students were requested to fill in a questionnaire, which included evaluating their competences in the related subjects before and after the summer school. Points were given between 1 (minimum) and 5 (maximum). The average result for the 20 students can be seen in Table II. The first seven competences are those more related to the competitions.

The students perception of their competences shows a mean value of 2.30 before the summer school, and 3.54 afterwards. This represents an increment of 1.24 (i.e., about 25%). Furthermore, this increment is with respect to initial good marks before the courses, since the students were already interested in the field: they have volunteered to participate in the Summer School, and their curricula already included subjects related to robotics and mechatronics. Interestingly, the increments for the competences with a special weight for the competition show less dispersion than the rest of the courses, since they are uniformly increasing by about 1.2 points. This might mean that the competition motivates the students beyond their particular initial interests, and that this

TABLE II
EVALUATION OF SUMMER SCHOOL COMPETENCES

MEAN for all competences	2.30	3.54	1.24
Competence	**Previous**	**After**	**Increment**
Using LabVIEW as a development tool	2.15	3.8	1.65
Understanding the basic problems in Mobile Robotics	2.85	4.05	1.2
Designing robotic control architectures	2.2	3.4	1.2
Testing and evaluating control architectures	2.5	3.45	0.95
Integrating sensors and actuators in real applications	2.6	3.8	1.2
Understanding Fuzzy Logic basics	2.55	3.75	1.2
Integrating Fuzzy Logic into control architecture	2.05	3.15	1.1
Understanding and using modern processor architectures including parallelism	2.1	3.5	1.4
Designing low-level hardware implementations using CAE-Design tools	2.15	2.95	0.8
Understanding and using Fourier transforms for image processing and visual navigation	2.65	3.85	1.2
Understanding algorithm parallelization and mapping	1.8	3.4	1.6
Using the internet to tele-operate remote systems	2.1	3.4	1.3

is independent of the difficulty and temporal arrangement of the courses, since these are enhanced by the ones of the following days. Finally, it is also remarkable that the best competence improvement (i.e., 1.65) was with LabVIEW, despite that for many students this was their very first contact.

CONCLUSIONS

The paper reports on the experience of the 2nd Summer School of Mechatronics, jointly organized by the Technical University of Dresden and the University of Málaga. The courses, with a duration of two weeks, were offered to advanced undergraduate students from both universities.

An important part of the lab-work and two student competitions have been based on the LEGO Mindstorms NXT Education Base Set. Instead of the default NXT Education Software, we have used LabVIEW with the NXT add-on, a general-use development platform directly supported by the NXT unit. The combined use of LabVIEW and LEGO kits allows hands-on lab work with real hardware and representative software.

Moreover, the paper proposes and describes basic LabVIEW structures for several case study laboratory practices. These can be provided as a model to students that are not proficient with this programming philosophy. Particularly, the two alternative programming models supported by the NXT add-ons have been addressed: NXT Toolkit, for stand-alone performance, and NXT Direct, for more powerful solutions run on the PC.

An assessment of student competences has been based on a survey. The overall increment for the Summer School competences has been of about 25%. Moreover, we found that the increments for the competences specially related to the competition show less dispersion than the rest of the courses.

By spending one week in each country with teams made of a student from each university, the summer school has gained in cultural dimension and knowledge sinergy.

Work for future editions will include enriching the competition with competences from a greater number of courses.

ACKNOWLEDGMENTS

The 2nd Summer Course on Mechatronics was partially funded by the University of Málaga and the Technical University of Dresden. Additional funding was obtained from TEP-1379 and TEP-375 (Junta de Andalucía, Spain). The authors are grateful to the rest of the teachers in the 2008 Summer School on Mechatronics: PD Dr.-Ing. Annerose Braune, Prof. Dr.-Ing. habil. Renate Merker, Prof. Dr.-Ing. habil, René Schüffny, from the Technical University of Dresden; and Prof. Dr. Nicolás Guil-Mata and Prof. Dr. Óscar Plata-González, from the University of Málaga.

REFERENCES

[1] M. Habib, "Mechatronics - a unifying interdisciplinary and intelligent engineering science paradigm," *Industrial Electronics Magazine*, vol. 1, no. 2, pp. 12–24, 2007.

[2] S. Waks, "Lateral thinking and technology education," *Journal of Science Education and Technology*, vol. 6, no. 4, pp. 245–255, 1997.

[3] J. González and R. Wagenaar, *Tuning Educational Structures in Europe, Phase 2*. Universidad de Deusto, 2005.

[4] M. Barak and Y. Zadok, "Robotics projects and learning concepts in science, technology and problem solving," *International Journal of Technology and Design Education*, 2007.

[5] E. Wang, "Teaching freshmen design, creativity and programming with LEGOs and Labview," in *Proc on 31st Annual Frontiers in Education Conference*, vol. 3, Nagoya, Japan, 2001, pp. 11–15.

[6] V. Papadimitriou and E. Papadopoulos, "Putting low-cost commercial robotics components to the test - development of an educational mechatronics/robotics platform using LEGO components," *Robotics and Automation Magazine*, vol. 14, no. 3, pp. 99–110, 2007.

[7] B. van der Vlist, R. van de Westelaken, C. Bartneck, J. Hu, R. Ahn, E. Barakova, F. Delbressine, and L. Feijs, *Technologies for E-Learning and Digital Entertainment*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2008, vol. 5093/2008, ch. Teaching Machine Learning to Design Students, pp. 206–217.

[8] M. Higa, D. Tawy, and S. Lord, "An introduction to LabVIEW exercise for an electronics class," in *Proc on 32nd Annual Frontiers in Education Conference*, vol. 1, Nagoya, Japan, 2002, pp. T1D13–T1D16.

[9] F. Adamo, F. Attivissimo, G. Cavone, and N. Giaquinto, "SCADA/HMI systems in advanced educational courses," *IEEE Transactions on Instrumentation and Measurement*, vol. 56, pp. 4–10, 2007.

[10] J. Gómez-de-Gabriel, "LabVIEW como herramienta de enseñanza en cursos de ingeniería remotos (in spanish)," Universidad de Málaga. Available online at: http://sine.ni.com/cs/app/doc/p/id/cs-11347, Tech. Rep., 2008.

[11] A. Leva and F. Donida, "Multifunctional remote laboratory for education in automatic control: The CrAutoLab experience," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 6, pp. 2376–2385, 2008.

[12] A. García-Cerezo, A. Mandow, J. L. Martínez, J. Gómez-de-Gabriel, J. Morales, A. Cruz, A. Reina, and J. Serón, "Development of ALACRANE: A mobile robotic assistance for exploration and rescue missions," in *IEEE International Workshop on Safety, Security, and Rescue Robotics (SSRR2007)*, Rome, Italy, 2007.