Dealing with Conflicting Requirements in Robot System Engineering: A Laboratory-Based Course

Luca Gherardi[1], Davide Brugali[2], and Andrea Luzzana[2]

[1] ETH Zürich, Switzerland
`lucagh@ethz.ch`
[2] University of Bergamo, Bergamo, Italy
{`andrea.luzzana,brugali`}`@unibg.it`

Abstract. This paper presents a project-based laboratory for senior-level students in computer engineering that is based on the LEGO Mindstorms kits extended with a set of off-the-shelf microcontrollers and custom electronics. It is organized in an integrated set of projects, which individually cover a subset of typical issues and challenges involved in the development of a complete robotic system. The pedagogical goal is to equip students with an understanding of how engineering of complex projects is a multi-dimensional decision making process and with teamwork and self-learning skills.

1 Introduction

Robustness, versatility, low-cost, performance, and reusability are examples of conflicting requirements that make the process of engineering robotic systems a difficult and challenging endeavour [3]. Robotic system engineers should master highly heterogeneous technologies in order to exploit and integrate them in a consistent and effective way. Thus, from an educational point of view, robot system engineering is both a challenge and an opportunity [10].

Teaching robotic system engineering is challenging because Robotics is an experimental science that plays the role of integrator of the most advanced results in a large variety of research fields and thus is highly dependant on the evolution of the underlying technologies. Teaching robotic system engineering should therefore focus on providing students with the skills (1) to identify stable and varying aspects in the domain of robotic systems, (2) to analyze conflicting requirements arising from the need to exploit and integrate heterogeneous technologies, and (3) to perform careful multidimensional modelling and design of complex systems where properties are emerging from the interaction of constituent parts.

Learning robotic system engineering is an opportunity to discover how theoretical concepts in a variety of scientific disciplines typically learned in different classes can be applied in practice, and how synergies among disparate technological fields can be exploited to build complex systems [2].

Aim of this paper is to present a project-based laboratory that senior-level students in computer engineering take before graduation at the Computer science Department at the University of Bergamo. It is an optional laboratory that follows a compulsory half-year course in Robotics.

D. Brugali et al. (Eds.): SIMPAR 2014, LNAI 8810, pp. 554–565, 2014.

The pedagogical goal of this laboratory is to equip students with an understanding of how engineering of complex systems is a multi-dimensional decision making process, which consists in analyzing and eliciting conflicting requirements, identifying alternative designs, selecting, implementing, and verifying tradeoff solutions, and how complexity incorporates not only technological issues, but also the human organization.

For this purpose, the laboratory has been structured as a Problem-Based Learning (PBL) course, where students are assigned an open-ended engineering problem [9], which: a) requires more information than is initially available, b) admits multiple solution paths, c) changes as new information is obtained, d) requires collaboration among students. The laboratory described in this paper has several elements of novelty compared with the state of the art.

First, it covers a larger set of topics than other project-based courses (e.g. [7]), as it allows students to face an integrated set of challenges related to mechanical design, wireless communication protocol design, motor control, sensor data processing, microcontroller programming, and PC programming. This is highly appreciated by students since the curriculum in computer engineering at the University of Bergamo can be customized by including courses in mechatronics, and mechanical engineering.

Second, it addresses the various phases of the robot engineering process, from requirements elicitation and analysis, to system design and subsystem development, up to system integration and validation. A similar approach has been documented in [4], where a course in design and implementation of a small robot is described. The small robot is much simpler than the kind of robotic system developed during the project-based laboratory described in this paper.

Third, it uses the LEGO kit not for its simplicity as in [12], [5], [11], and [6] but for its versatility [13]. Indeed, the LEGO RCX has been replaced by a more powerful low cost microprocessor in order to control a larger number of motors and sensors than it is allowed by the RCX or the NXT devices.

Fourth, it is organized in an integrated set of projects, which individually cover the issues and challenges involved in the development of a specific subsystem of the complete robotic system. Each project is assigned to a small group of students, who have to complete their assignment taking into account the requirements of their subsystem and the constraints imposed by the other subsystems. This organization allows students to learn the importance of proper documentation of project results both as users and providers. In contrast, the courses described in the literature (e.g. [14], [6]) are typically organized as a set of simple and independent projects.

The paper is organized as follows. Section 2 summarizes the curriculum in computer science offered at the University of Bergamo. Section 3 presents the laboratory assignment and describes the LEGO robotic system that has been developed during the laboratory. Section 4 presents the organization of the laboratory in terms of student groups and activities. Section 5 illustrates the system engineering challenges faced by the students. Finally Section 6 reports on the lesson learned and on the project evaluation, and draws the relevant conclusions.

2 Course Description

The University of Bergamo offers a computer engineering degree organized in two levels (3 years and 2 years long). The project-based laboratory, presented in the paper is complementary to the course of Robotics, which is offered during the first semester of the last year of the second level degree, is mandatory in the Mechatronics and Industrial Informatics curricula and optional in the Networked Information Systems curriculum.

The objective of the Robotics course (9 CFUs)[1] is to provide an introduction to the fundamental concepts, models, and algorithms to develop software control systems for autonomous mobile manipulation robots. The key topics include: (a) robot kinematics, (b) motor control, (c) robot perception (laser, sonar, 2D-3D camera), (d) motion planning and navigation, (e) control and software architectures with a specific focus on reusing open source libraries. The course spans over 12 weeks in the first semester; it is made of lectures of 3 hours each, that are given twice a week for a total of 24 lectures.

Before the Robotics course, students follow several courses in computer science, control, electronics, and mechatronics, which provide the required background for the project-based laboratory described in this paper, such as: (a) high level programming languages (36 CFU), (b) embedded, real-time, and distributed system programming (15 CFU), (c) digital control and system identification (21 CFU), (d) multi-body systems modeling and design (6 CFU).

3 The LEGO Mobile Manipulator

The overall goal of the project-based laboratory, declared to the students during the first day, was the following. "*The final objective is the design and the realization of a mobile manipulator. The rover must be able to move towards a desired position (expressed in terms of x, y, θ with respect to the initial pose reference frame) while the arm to reach any pose in its 3D workspace (expressed in terms of joint positions). The human operator specifies the rover and arm target positions through a graphical user interface running on a standard PC. The effectiveness of the design should be evaluated in terms of* robustness *by defining a stress test for the hardware,* versatility *by analyzing the shape of the workspace, and* performance *by analyzing position accuracy and repeatability.*"

The assignment didn't specify any specific kinematic model, neither for the rover nor for the arm. The students were also free to decide the more appropriate localization mechanism to be used for computing the rover position (e.g. odometry, visual based), the number of computational nodes, and the distribution of functionality among the computational nodes.

The result of the laboratory-based course is the mobile manipulator robot depicted in Figure 1 (left). The robot is composed of an omnidirectional wheeled

---

[1] University Formative Credit (CFU):1 CFU correspond to 25 hours of study including homeworks.

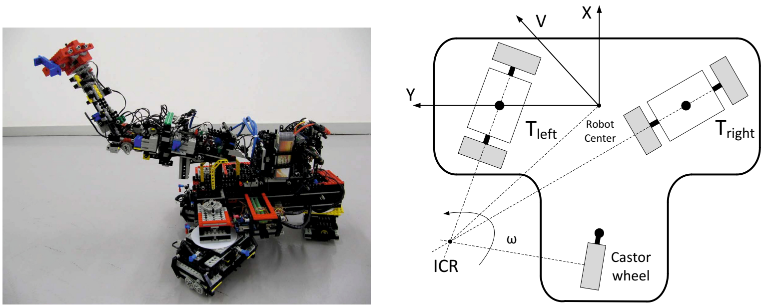

Fig. 1. The mobile manipulator (left) and the rover kinematic model (right)

rover and a 6 degrees of freedom (DOF) arm. The rover and the arm are controlled by two onboard microcontrollers that communicate with a remote PC workstation. The workstation executes a GUI that allows the user to specify a target position of the arm joints and a target position of the rover with respect to a global reference frame. A simple navigation algorithm localizes the robot using images captured by a ceiling camera, computes the trajectory (turn on place and move forward) from the current position to the target position and sends velocity commands to the rover MCU.

The robot has been built using six LEGO Mindstorms kits. In total, ten DC motors (9V, 300 mA), twelve rotary encoders with a resolution of 16 steps per revolution, and eight contact sensors were used. The motors allow a no-load maximum rotation speed of 360 rpm and a stall torque of 5.5 Ncm.

The LEGO RCX computational unit has been replaced by a more powerful STR32 microcontroller (MCU). The MCU interfaces the LEGO motors and sensors through custom electronic boards developed by the students during another course project. All the devices are power supplied by 3 cells LiPo batteries with a nominal voltage of 11.1 V and 2 A/h capacity. The MCU communicates with a remote PC over an 802.15.4 wireless network provided by Maxstream ZigBee modules [8] which establish a broadcast wireless connections offering a bitrate of 115200 bit/s.

3.1 The Robot Kinematics

The kinematic structure of the omnidirectional nonholonomic rover (see right part of Figure 1) is based on two separate differential drive subsystems (T_{left} and T_{right}), linked to a rigid platform by two passive steering axis, and a castor wheel. Each subsystem has a couple of actuated wheels coupled with a rotary encoder providing speed feedback and a third rotary encoder measuring the angular position of the steering axis.

The desired motion of the robot is specified by the linear velocity vector V and the rotational speed ω expressed in the robot reference frame (X, Y). These two parameters identify an unique instantaneous center of rotation (ICR) around which the robot will move. When the ICR position changes, the wheels of each differential drive subsystem are actuated with opposite velocities in order to rotate it on place so that the wheels axes intersect at the new ICR. Then, each wheel is actuated in order to reach a reference speed.

The kinematic structure of the robotic arm is that of a typical 7-DOF robotic arm, where the first axis is replaced by the underlying rover. The last three joints intersect in a single point (wrist center) in order to satisfy the Pieper condition and simplify the closed form solution of the inverse kinematics.

Each one of the six joints is actuated by a LEGO motor and mounts a rotary encoder to measure the angular position. A contact sensor, acting as a limit switch, is used to obtain the home position of the joint at startup. The arm MCU executes six separated closed loop position controllers.

4 Laboratory Organization

The objective of the project-based laboratory was to allow students to face the challenge of developing a complex system. Here, according to the etymology, complex does not mean complicated but interlaced. Indeed, the development of the LEGO robot described in Section 3 provides food for thought along two interlaced dimensions: spatial and temporal.

The spatial dimension is concerned with the modular structure of the robotic system (the rover, the arm, the onboard and the offboard computation). Considered the number of students, the development of the entire robot was broken down by the instructors into four projects (depicted in Figure 2), which were defined according to following principle: the projects (a) had to lead to the development of composable building blocks, so that they could be carried out concurrently, and (b) had to be interdependent, so that they could stimulate the discussion and the interaction between the groups.

The first two projects were assigned to groups of three students with a specific interest in mechatronics, while the other two projects to groups of four students with an interest in industrial informatics or information systems. Each project spanned a total of twelve weeks. Students carried on their projects during the sessions attended by the tutors (four hours per week) and met in the laboratory at least two additional times per week.

The temporal dimension is concerned with the development process, which requires the students to analyze, disentangle, and negotiate conflicting requirements, to revise design decisions according to ongoing work by other students, and to integrate heterogeneous technologies. The projects were structured in four phases, according on the typical design stages of mechatronic projects [14].

During the first phase (*Requirement elicitation and Technology assessment*), the four groups were invited to internally discuss the project assignment, to devise the requirements for the subsystem to be developed, to discuss these requirements with the other groups, to survey the available literature, and to get the necessary software tools.

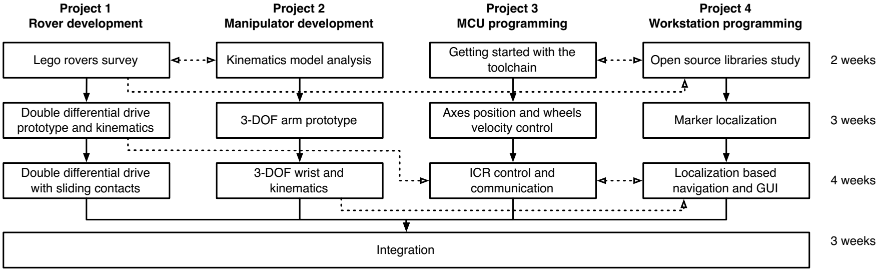

Fig. 2. The projects steps and dependencies (dashed arrows)

During the second and third phase (*Subsystem design and prototyping*), respectively, each group designed and developed a first prototype and the final version of its subsystem.

The fourth phase (*System integration and evaluation*) was for the integration of the system. It required the groups to coordinate their efforts in order to solve any inconsistencies in the software and in the hardware.

While the organization of the spatial dimension was fixed, the students had the possibility to discuss with the instructors the organization of activities carried out during the four development phases defined along the temporal dimension.

5 System Engineering Challenges

The students of the four projects faced several development challenges that allowed them to realize the complexity of system engineering and to learn the importance of negotiating conflicting requirements. The challenges can be grouped according to the development phases: requirement analysis, design and implementation, and integration. They are described in the next three subsections.

5.1 Challenge 1: Developing Feasible Requirements

During the first phase, the students faced the issue of analyzing the project assignment and eliciting the required functionality. The project assignment reflects a typical problem in system engineering: the customer needs are normally described as a wish list. In particular, the project assignment generically indicates that the robot has to be a mobile manipulator and that it has to be able to reach a pose in the 3D environment specified by the user.

The students of Project 1 analyzed different kinds of LEGO rovers documented in literature (such as [5]). The ambition of creating a kind of LEGO rover never developed before motivated the students to design an omnidirectional robot. Furthermore, the ominidirectional kinematics model would greatly satisfy the *versatility* requirement specified in the project assignment.

After a survey of the literature, the students quickly realized that omnidirectional rovers typically use Swedish wheels, which are not included in the LEGO

Mindstorms kit and cannot be easily built using LEGO bricks. They understood that in this case *Versatility* and *Reusability* are conflicting requirements.

Similarly, the students of Project 2 surveyed the literature on manipulator kinematic models. In order to satisfy the *Versatility* requirement, they evaluated the possibility to build a 7-DOF arm such as the Kuka LWR (3 DOF for the shoulder, 1 DOF for the elbow, and 3 DOF for the wrist). The students realized that such a robot would be difficult to build with LEGO bricks and moreover it would be an excessive weight for the rover. They understood that in this case *Versatility* and *Robustness* are conflicting requirements.

The two groups understood that the kinematics of the mobile manipulator had to be specified jointly. A viable alternative for building an omnidirectional rover has been identified by taking inspiration from [1]: the students of Project 1 decided to design a double differential drive rover, which is feasible reusing LEGO bricks and provides both stability and traction power (four motors for traction and steering). In order to take into account the payload limitations of the rover, the students of Project 2 decided to develop an arm with only 2 DOF for the shoulder since the missing DOF is provided by the rover.

The two groups discussed this choice with the students of Project 3, who emphasized the complexity of developing the control software for the double differential drive due to the need to synchronize and coordinate the motion of the two traction systems. They also pointed out that this issue would have affected the performance (position accuracy and repeatability) of both the rover and the arm. Here students understood that in this case *Robustness* and *Performance* are conflicting requirements.

The students of Project 1 raised a concern about the possibility to estimate the rover motion using only odometry due to the limited resolution of the encoders and the difficulty in controlling the double-differential rover. Two solutions were considered: using an onboard webcam to track visual markers placed on the floor in known positions or mounting the webcam on the ceiling in order to track a visual marker on the rover. Both solutions would satisfy the *Performance* requirement. The former solution would better satisfy the *Versatility* requirement, since the workspace would not be limited by the field of view of the camera mounted on the ceiling, but was immediately discarded because the students of Project 3 pointed out the limitations of the onboard MCU, which does not have enough power to process camera images. Here students understood the importance of *Resource constraints* in the design of embedded systems.

Students of Project 3 discussed with their colleagues of Project 4 about how to distribute the functionality of the robot between the MCUs and the workstation taking into account the low bitrate of the wireless communication and the different characteristics of the computational units. The MCU can perform real-time tasks by reacting to hardware interrupts in a very short time but has a limited computational power. On the contrary, the workstation PC has a high-frequency CPU but no efficient I/O mechanisms. Thus, the students agreed to implement the wheels closed-loop speed controller and ICR controller on the rover MCU, the joints closed-loop position controller on the arm MCU, and the image processing and GUI on the workstation.

5.2 Challenge 2: Integrating Heterogeneous Technologies

During the second and third phase, the students faced the issue of designing and implementing the four subsystems. In particular, they realized that the LEGO Mobile Manipulator is a good example of heterogeneous structures where properties (*Robustness*, *Versatility*, *Performance*) are emerging from the interaction of constituent parts and cannot be confined into individual subsystems.

The students of Project 2 faced several design issues due to the limited resolution of the encoders and the gears backlash. In particular they had to consider two conflicting requirements: position accuracy and repeatability. The former can be achieved by mounting the encoder between the motor shaft and the reduction gearbox of the joint. The latter, which is more important in industrial applications, can be achieved by mounting the encoder between the reduction gearbox and the joint shaft. The students understood that the performance of the robotic arm results from the specific integration of the mechanical subsystem and the sensor subsystem. Indeed, they realized that the position uncertainty due to the limited resolution of the encoder is higher than the gear backlash, thus repeatability could not be improved by mounting the encoder on the joint axis, while it was much easier to mount it on the motor axis.

Students of Project 2 realized that it was crucial to place the three motors of the wrist as much as possible close to the base of the arm in order to have a lower weight on the joints. This requirement could be met by building a complex differential transmission gearbox. As a result the joints of the wrist couldn't be moved independently requiring the definition of the transformation from joints velocity to speed of the motors. The students of Project 2 and Project 3 understood that systems engineering typically involves design decisions, whose effects are not local to individual subsystems but span over interconnected systems. Indeed, the design of the differential transmission implied a significant higher effort to implement the axis controllers on the MCU of the arm.

The students of Project 3 and Project 4 discussed the specification of a shared serial communication protocol between the MCUs and the remote workstation. They agreed on the packet length and structure, the commands identifiers and parameters, and the units of measurement of the exchanged data. The students understood the importance of separating the common interface between two interdependent functionalities from their specific implementations. Indeed, the students of the two groups could focus on the implementation of the functionality for motion control on the MCU and for navigation on the workstation independently. In particular, the students of Project 3 implemented the functions to read the encoders, to generate the PWM output for the motors, and to read the serial communication peripheral. The students of Project 4 implemented a simple GUI for sending commands to the rover and to the arm, using software libraries learned in previous courses. They also implemented a simple navigation algorithm that periodically localizes the rover using the ARTK+ library, computes the straight line between the rover current position and the target position, and generates the velocity commands to turn the rover on place and to drive it toward the target.

5.3 Challenge 3: Revising Requirements and Design Decisions

The development of the LEGO Mobile Manipulator has been a highly iterative process as is typical in complex systems engineering. The students had to solve some design problems that emerged only when they started testing and integrating the various subsystems.

The first prototype of the double differential drive had a notable limitation: the traction subsystems weren't able to steer more than 180 degrees because of the cables used for transmitting signals and power between the MCU, the motors and the encoders. Thus the students considered the possibility to improve the prototype in order to remove the steering limitation. They understood that the interface between the mechanical subsystem and the electronic subsystem was not well defined. They realized that a sliding contact along the turning axis of each traction system was needed. They faced here two conflicting requirements: *Performance* and *Reusability*, i.e they had to increase the steering capability of the rover using material available in the laboratory. The chosen approach was to build two sliding contacts using only LEGO bricks and copper wires. The rotor is made up of eight coaxial pulleys mounted on the revolving axis. Eight copper cables are rounded on each pulley and come out from the bottom of the rotor in order to be connected to two motors and two encoders. The stator is made up of sixteen coaxial vertical supports, which tense eight copper cables around the pulleys. These cables come out from the top of the stator in order to be connected to the MCU.

Once assembled, the LEGO mobile manipulator robot has been tested in order to validate the overall system and identify design errors. The following problems have been identified and solved during this phase:

- The load of the rover was higher than expected and not well balanced. As a consequence, the castor wheel was not able to turn adequately when the rover had to change direction. This problem has been mitigated by better distributing the load of the batteries.
- Oscillations in the motors movements led to a flickering motion of the rover. This problem has been addressed by better tuning the PI parameters of the wheel speed controllers.
- Several data packets were transmitted incorrectly during the communication between the workstation and the two MCUs on the robot. Students realized that the wireless communication in broadcast mode was unreliable and decided to implement a protocol that checks for corrupted packets in order to retransmit them.
- The marker localization algorithm was highly sensitive to the scene illumination due to the limited capability of the low-cost webcam. They suggested to use an additional light source to improve the scene illumination.

By integrating their subsystems and evaluating the resulting system, students had an additional opportunity to learn about the constraints that characterize systems engineering.

6 Evaluation and Concluding Remarks

Students were required to submit a technical documentation of their achievements at the end of each phase using the Trello (`http://trello.com`) collaboration tool, which provided each group of students with an individual project board and each student with an individual account. This system allowed the instructor to track the contribution of each student to the project development and assess the student learning (50% of the score). The documentation has been evaluated for completeness and adequacy of the bibliographic references and for the accuracy of the technical description of the proposed solution (e.g. the use of standard modeling languages, such as UML for documenting the software). The rest of the grade was based on oral questions designed to check to what extent students contributed to the project, their role in the group, and their understanding of the system requirements, the motivations underlying the design choices, and the correlation between design choices and system behaviour during the experimental evaluation of the robotic system functionality.

The analysis of the documentation and the oral exam clearly indicates that the students learned three fundamental lessons emerging from the challenges they faced during the development of the LEGO mobile manipulator.

The first lesson learned is that a careful analysis of the informal user's requirements (i.e. project assignment) and of the technical specifications (i.e. LEGO kit) must be performed in order to develop a good set of feasible requirements taking into account the resources available to the project (material, time, knowledge).

The second lesson learned is that robotic systems engineering is inherently complex due to the interdisciplinary skills required, the heterogeneous technologies involved, and the difficulty in characterizing the interactions among systems and subsystems.

The third lesson learned is that developing robotic systems is a highly iterative process that may require to revise initial requirements and design decisions. Even if the initial problem has been carefully decomposed in subproblems (the four projects), their individual solutions (the four subsystems) may not fit together particularly well at first.

The project-based laboratory has been introduced for the first time in the 2008-2009 session as complementary to the Robotic course. Beside the specific skills acquired through the project development, the positive effects of the laboratory on the student learning achievement can be measured indirectly by observing their grades attained for the Robotics course in each session.

As indicated in Table 1, in contrast to previous editions of the Robotics course, more students have passed the exam right after the end of the course and the average grades were higher. These results indicates that the project-based laboratory allowed the students to gain a better understanding of the theoretical

Table 1. Student attainment of the Robotics course before and after the introduction of the project-based laboratory

Metric	2001-2008	2009-2013
Mean success rate (%)	73	87
Mean score (min 18, max 30)	25	27

concepts presented in the Robotics course. In addition the instructors observed a more regular and involved students' partecipation during the classes.

With regard to the pedagogical objectives illustrated in Section 1, the development of LEGO robot has presented both strengths and weaknesses.

The proposed project-based laboratory offered the students the opportunity to appreciate the multidisciplinary nature of robotics, and to investigate the close relationship between software and hardware design. It is easily scalable and can be offered to larger classes. For example, a fifth group of students could develop a perception system, which uses the LEGO light sensor to drive the rover along a visual path on the floor. A sixth group of students could use the same sensor to recognize colored spots on a wall. In this case, the sensor could be mounted on the arm. In both cases, additional conflicting requirements would emerge.

At the same time, the limitations of the LEGO kit generated a sense of frustration in the students, who got excited about creating an entire robot from scratch but got disappointed about the high technological gap between the project-based laboratory and other theoretical courses in their curriculum. More specifically, the groups who developed the rover and the arm used a trial-and-error method of direct implementation of the chosen kinematics model since the mechanical properties of LEGO bricks (gears, wheels, pulley, joints) were not available for performing a model-based design and evaluation of transmission efficiency, backlash, and wear as they learned in previous courses. In contrast, the students who programmed the MCU followed the Ziegler-Nichols empirical method to tune the PI parameters of the position and speed controllers

The validity of the approach has been evaluated by requesting the students to fill an anonymous questionnaire, which is common to all the courses of the Faculty of Engineering at the University of Bergamo. Table 2 summarizes the results, which reports the mean and the spread of the scores for the project-based laboratory presented in this paper and the mean of the scores for all the courses of the Faculty of Engineering (the maximum score was ten). Overall the project-based laboratory has been successful, as demonstrated by the high scores of the first three questions. Students found it appropriate to have this course in their curriculum, they were highly motivated to contribute to the success of the team work, and considered the topic very interesting. The workload has been perceived to be in line with other courses. Most students found unusual the lack of a textbook and the need to search for adequate material by themselves.

Table 2. Scores of the evaluation questionnaires

Question	**Mean**	**Spread**	**Mean (Faculty)**
Curriculum Organization	7.9	1.5	7.4
Motivation	8.4	0.7	7.3
Interest in the topic	8.1	1.4	7.2
Workload adequacy	7.1	1.2	7.0
Teaching material adequacy	7.0	2.7	8.0
Assessment Method	7.2	1.7	7.9

References

1. Borenstein, J.: Experimental results from internal odometry error correction with the omnimate mobile robot. IEEE Transactions on Robotics and Automation 14(6), 963–969 (1998)
2. Brugali, D.: Exploiting the synergies between robotics and software engineering: A project-based laboratory. In: Global Engineering Education Conference (EDUCON). IEEE (2014)
3. Brugali, D., Gherardi, L., Biziak, A., Luzzana, A., Zakharov, A.: A reuse-oriented development process for component-based robotic systems. In: Noda, I., Ando, N., Brugali, D., Kuffner, J.J. (eds.) SIMPAR 2012. LNCS (LNAI), vol. 7628, pp. 361–374. Springer, Heidelberg (2012)
4. Crisman, J.: System design via small mobile robots. IEEE Trans. Educ. (1996)
5. Kim, S., Jeon, J.: Introduction for freshmen to embedded systems using lego mindstorms. IEEE Trans. Educ. 52(1), 99–108 (2009)
6. Kim, Y.: Control systems lab using a lego mindstorms nxt motor system. IEEE Trans. Educ. 54(3), 452–461 (2011)
7. Lee, C., Su, J., Lin, K., Chang, J., Lin, G.: A project-based laboratory for learning embedded system design with industry support. IEEE Trans. Educ. (2010)
8. Maxstream XBee ZigBee, `http://www.digi.com/`
9. O'Grady, M.J.: Practical problem-based learning in computing education. Trans. Comput. Educ. 12(3), 10:1–10:16 (2012)
10. Padir, T., Chernova, S.: Guest editorial special issue on robotics education. IEEE Transactions on Education 56(1), 1–2 (2013)
11. Papadimitriou, V., Papadopoulos, E.: Putting Low-Cost Commercial Robotics Component to the Test. IEEE Robotics & Automation Magazine 14, 99–110 (2007)
12. Ruiz del Solar, J.: Robotics-centered outreach activities: An integrated approach. IEEE Trans. Educ. 53(1), 38–45 (2010)
13. Ruzzenente, M., Koo, M., Nielsen, K., Grespan, L., Fiorini, P.: A review of robotics kits for tertiary education. In: Proc. Int. Workshop Teaching Robot. Teaching Robot., Integr. Robot. School Curric., Riva del Garda, Italy (2012)
14. Tutunji, T., Saleem, A., Rabbo, S.A.: An undergraduate mechatronics project class at philadelphia university, Jordan: Methodology and experience. IEEE Trans. Educ. 52(3), 365–374 (2009)