



MATLAB Meets LEGO Mindstorms – A Freshman Introduction Course Into Practical Engineering

A. Behrens¹, L. Atorf¹, R. Schwann², B. Neumann², R. Schnitzler², J.
Ballé³, T. Herold⁴, A. Telle⁵, T.G. Noll², K. Hameyer⁴, T. Aach¹

¹Institute of Imaging and Computer Vision, ²Institute of Electrical Engineering and
Computer Systems, ³Institute of Communications Engineering,
⁴Institute of Electrical Machines and ⁵Institute of Communication
Systems and Data Processing

RWTH Aachen University, 52056 Aachen, Germany
tel: +49 241 80 27860, fax: +49 241 80 22200
web: www.lfb.rwth-aachen.de

in: Education, IEEE Transactions on. See also BIB_{TEX} entry below.

BIB_{TEX} :

```
@ARTICLE{Behrens2010e,  
author={Behrens, A. and Atorf, L. and Schwann, R. and Neumann, B. and Schnitzler, R. and Balle, J.  
and Herold, T. and Telle, A. and Noll, T. G. and Hameyer, K. and Aach, T.},  
journal={Education, IEEE Transactions on},  
title={MATLAB Meets LEGO Mindstorms -- A Freshman Introduction Course Into Practical Engineering},  
year={2010},  
month={May},  
volume={53},  
number={2},  
pages={306--317},  
keywords={},  
doi={10.1109/TE.2009.2017272},  
ISSN={0018-9359}  
}
```

© copyright by the author(s)

MATLAB Meets LEGO Mindstorms—A Freshman Introduction Course Into Practical Engineering

Alexander Behrens, *Student Member, IEEE*, Linus Atorf, Robert Schwann, Bernd Neumann, Rainer Schnitzler, Johannes Ballé, Thomas Herold, Aulis Telle, Tobias G. Noll, *Senior Member, IEEE*, Kay Hameyer, *Senior Member, IEEE*, and Til Aach, *Senior Member, IEEE*

Abstract—In today’s teaching and learning approaches for first-semester students, practical courses more and more often complement traditional theoretical lectures. This practical element allows an early insight into the real world of engineering, augments student motivation, and enables students to acquire soft skills early. This paper describes a new freshman introduction course into practical engineering, which has been established within the Bachelor of Science curriculum of Electrical Engineering and Information Technology of RWTH Aachen University, Germany. The course is organized as an eight-day, full-time block laboratory for over 300 freshman students, who were supervised by more than 60 tutors from 23 institutes of the Electrical Engineering Department. Based on a threefold learning concept comprising mathematical methods, MATLAB programming, and practical engineering, the students were required to transfer mathematical basics to algorithms in MATLAB in order to control LEGO Mindstorms robots. Toward this end, a new toolbox, called the “RWTH—Mindstorms NXT Toolbox,” was developed, which enables the robots to be controlled remotely via MATLAB from a host computer. This paper describes how the laboratory course is organized and how it induces students to think as actual engineers would in solving real-world tasks with limited resources. Evaluation results show that the project improves the students’ MATLAB programming skills, enhances motivation, and enables a peer learning process.

Index Terms—Freshman course, introduction to programming, LEGO mindstorms, MATLAB, practical engineering, project-based learning, robotic laboratory.

Manuscript received July 10, 2008; revised February 16, 2009. First published August 18, 2009; current version published May 05, 2010.

A. Behrens, L. Atorf, and T. Aach are with the Institute of Imaging and Computer Vision, Faculty of Electrical Engineering and Information Technology, RWTH Aachen University, Aachen 52062, Germany (e-mail: behrens@lfb.rwth-aachen.de; aach@lfb.rwth-aachen.de).

R. Schwann, R. Schnitzler, and T. G. Noll are with the Institute of Electrical Engineering and Computer Systems, Faculty of Electrical Engineering and Information Technology, RWTH Aachen University, Aachen 52062, Germany.

B. Neumann is with Automotive Electronics, Robert Bosch GmbH, Reutlingen 72703, Germany.

J. Ballé is with the Institute of Communications Engineering, Faculty of Electrical Engineering and Information Technology, RWTH Aachen University, Aachen 52062, Germany.

T. Herold and K. Hameyer are with the Institute of Electrical Machines, Faculty of Electrical Engineering and Information Technology, RWTH Aachen University, Aachen 52062, Germany.

A. Telle is with the Institute of Communications Systems and Data Processing, Faculty of Electrical Engineering and Information Technology, RWTH Aachen University, Aachen 52062, Germany.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TE.2009.2017272

I. INTRODUCTION

TRADITIONAL undergraduate courses in electrical engineering often deal with mathematical and physical basics, principles of electrical engineering, and computer science from a theoretical point of view. First, the theoretical foundations are taught, and only then are the more mature students confronted with genuine engineering problems. In their first semesters, many students thus find it difficult to relate the theoretical concepts to practical engineering tasks, and consequently, their motivation drops [1].

Therefore, various innovations in electrical engineering curricula have been introduced during the last 15 years to prevent this motivation drop in the first semesters. The key to successful and effective learning was stated by Saint-Nom and Jacoby as “no matter how many concepts you teach, no matter how deep you go, no matter how tough your exams are, the only lessons that will remain in the students’ minds are those that touch them” [2, p. 545]. Thus, a pedagogical approach that places students in situations where they “feel like engineers” is likely to enhance student motivation best [3]. Consequently, teaching engineering skills becomes an important part of undergraduate lectures [4]–[6]. Establishing project-based laboratories using interactive software tools [7]–[9] as well as robot kits [3], [10]–[19] to teach core electrical engineering and computing concepts are becoming more and more common in current curricula. Furthermore, project work also often helps students to acquire social and professional skills in teamwork, management, communication, and presentation [2], [3], [5], [6], [17], [18], [20]. Some students may initially view such project challenges as more demanding than traditional lectures in which their own role is more passive than it is in project-based courses. However, active discussions with peers and supervisors soon provide other perspectives, and project-based lectures turn out to be more effective [13], [21], [22].

An additional aspect is that basic circuit theory courses commonly focus on analog electronics about which, in today’s era of digital systems, fewer students feel enthusiastic than was the case one or two decades ago [23]. At various universities, students are therefore introduced to discrete-time signals and systems right at the beginning of the curriculum. A pioneering textbook, suitable for freshman students, is [24]. Teaching mathematics and digital signal processing (DSP) by presenting and analyzing digital systems strengthens the theoretical foundations for subsequent electrical engineering subjects and

complements the classical subjects of analog technology [21], [23]–[26].

For computer-assisted teaching of mathematical and digital signal processing basics, mathematical programming tools like MATLAB from The MathWorks [27] or LabVIEW from National Instruments, which are also widely used in industry [23], [26], can be employed. Since text-based and graphically oriented tools led to similar results in freshman programming courses [28], both programming tools are about equally well suited.

Thus, a first-semester learning module combining digital signal processing with computer programming and problem-oriented engineering gives students their first insights into practical methods and basic engineering concepts and helps them to apply their knowledge to other challenges later on in their studies. With the introduction of the new Bachelor of Science curriculum in Electrical Engineering and Information Technology at RWTH Aachen University, Aachen, Germany, as stipulated Europe-wide by the European Bologna Process [29], a new first-semester project, called “MATLAB meets LEGO Mindstorms” was established in the academic year 2007–2008. This project-based laboratory is closely tied to a first-semester lecture on linear algebra and digital signal processing and focuses on three major objectives: applying mathematical methods and digital signal processing fundamentals to practical applications, teaching programming basics and introducing MATLAB as a tool for practical engineering, and emulating practical engineering tasks with LEGO Mindstorms NXT robots [30].

This paper describes the project’s aims, its contents, and its implementation to train 309 freshman students. Also, evaluation results obtained at the end of the project are given. It is organized as follows. Section II describes the learning targets of the laboratory. Then, the project structure and management are presented in Section III. Section IV introduces the new MATLAB toolbox that was designed for this project. The project is divided into mandatory basic exercises and a selection of more comprehensive ones, which are described in Section V. Evaluation results are given in Section VI. Section VII provides some future perspectives, and Section VIII gives the main conclusions about this practical course.

II. LEARNING TARGETS

The three main objectives of the “MATLAB meets LEGO Mindstorms” laboratory are to:

- 1) transfer mathematical concepts and methods to practical engineering tasks;
- 2) use MATLAB for the implementation of mathematical concepts on a computer;
- 3) combine MATLAB programs with practical engineering and real-world applications, which go beyond simulations and computer programming alone.

Thus, the content of the laboratory reflects the three key objectives *Mathematical Methods*, *MATLAB programming*, and *Practical Engineering*.

The mathematical fundamentals to be applied and expanded within the project were taught by an affiliated first-semester lecture called “Mathematical Methods of Electrical Engineering.” Besides providing a 2-h MATLAB introductory exercise, it covers discrete-time signals, digital signal processing, and system theory, all in a manner suitable to freshman students [24]. It includes topics such as definitions of signals and systems, sinusoids, complex numbers, Fourier analysis, sampling and aliasing, LSI systems and difference equations, FIR filtering (e.g., to smooth data), frequency response of FIR filters, polynomials, and z -transform of FIR filters. Also, MATLAB examples and demos are provided, and explicit links to the mathematics course on linear algebra, which runs in parallel, are given. Over the past decade, similar teaching experiences have already shown that such DSP introduction courses can be successfully taught to freshman students [23], [25], [26], especially when combined with computer-based projects.

In the laboratory, MATLAB is used for programming since it provides a direct and intuitive access to vectors and matrices. Moreover, this software is also widely used in industry and in advanced lectures at RWTH Aachen University. During the project, the students acquire MATLAB programming skills by translating mathematical algorithms to computer implementations. Even if some have not yet acquired the skills to realize their own ideas, the creation of MATLAB code to reproduce the given project exercises is important in their exploring their own capabilities [2].

The practical-engineering component of the laboratory goes beyond computer programming and is realized using programmable, remotely controlled robots. Other practical courses have already shown that robotics projects are even more attractive to students than computer programming alone and have the additional advantage of being more realistic than simulations [3], [10]–[19]. Students connect their use directly with technology and regard them as educational games [31]. A common approach to providing a robotics environment for teaching computer engineering and basic programming concepts is to employ programmable toy robot kits like LEGO Mindstorms [3], [10]–[12], [15], [18], [19], [31]–[35]. Many students are already familiar with LEGO, and the Mindstorms NXT system can be easily extended with additional features and sensors. This first-semester laboratory was therefore equipped with 100 LEGO Mindstorms NXT education sets [36]. The standard education kit provides a programmable NXT brick with an integrated USB and Bluetooth communication interface, four different types of sensors (touch, sound, light, and ultrasonic distance sensor), and three motors, as illustrated in Fig. 1. An advantage of using these robot kits is that, on the one hand, rather complex robots can be easily constructed, while on the other hand, the limited accuracy of the sensors and actuators as well as the firmware limitations of the NXT bricks pose constraints under which the project tasks need to be solved. Such limitations of project resources are essential in emulating authentic engineering tasks.

The laboratory includes six basic exercises and one major task, which may be selected individually. Each exercise is documented and organized in modular subtasks with clear requirements and targets. Each subtask addresses a different



Fig. 1. LEGO Mindstorms NXT hardware of the standard education kit: five sensors (light, sound, ultrasonic, and two touch sensors), three servo motors, and programmable NXT brick.

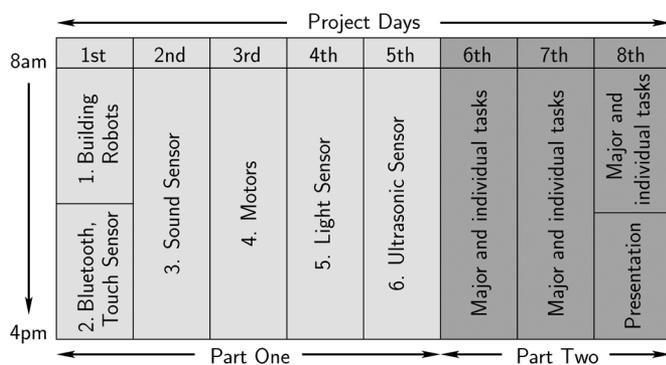


Fig. 2. Laboratory time schedule (in days). The eight-day full-time project is divided into two parts. In the first part, the students solve mandatory basic exercises, which address the NXT motors and different NXT sensors. The second project part includes major individual tasks and a final presentation.

NXT sensor or actuator, with a progressive level of difficulty from one subtask to the next. All the basic exercises are designed for students to work in pairs, learning programming principles in MATLAB, applying the RWTH—Mindstorms NXT Toolbox and the Bluetooth interface for computer–robot communication, exploring the NXT hardware characteristics, and constructing robots. The final major exercise gives the students—now grouped in teams of four—freedom to realize their own ideas, robot constructions, and inventions. For less creative students, a selection of three prepared major tasks with detailed descriptions can be provided on demand. In their solutions, the students have to take LEGO Mindstorms specifics into account, such as differences due to production spread between sensors or motors of the same type, and external conditions such as floor traction and random Bluetooth transmission delays. At the end of the laboratory, each student team has to present their individual robots, algorithms, and solutions. An overview of the whole laboratory time schedule divided into its two parts is given in Fig. 2.

Since the laboratory encourages the students to manage their own projects and gives them room to create individual robots and applications, student motivation may be expected to be higher than in classical courses. Moreover, the project fosters the peer learning process [20], [37], in which the students, within their teams, acquire soft skills like collaboration, teamwork, and leadership. Like professional engineers, the students have to solve problems in limited time with limited resources available and finally to present their approaches, solutions, and results in limited time slots.

Overall, the laboratory encourages freshmen to handle practical issues by building their own Mindstorms robots and applications and developing algorithms in MATLAB for signal processing and controlling purposes, based on mathematical methods.

III. PROJECT STRUCTURE AND MANAGEMENT

The project started in December 2007. It was carried out as an eight-day full-time block course, during which the students could focus entirely on the project while all other first-semester lectures paused. In the academic year 2007–2008, 309 freshmen participated in the laboratory. They were distributed over 23 institutes of the Electrical Engineering Department, where they were guided by more than 60 supervisors. In total, about 150 workstations and 100 LEGO Mindstorms NXT robot kits were provided.

When entering the Bachelor's program, the freshmen's programming skills differ strongly, ranging from excellent to literally nonexistent. During a two-month period before the project started, the accompanying lecture “Mathematical Methods of Electrical Engineering” provided a first MATLAB introduction lesson and short program examples with a focus on digital signal processing. In addition, a specially developed MATLAB Primer was supplied, which, apart from introducing students to MATLAB, also covered the required MATLAB basics for the laboratory. To allow the students to assess their MATLAB skills, a multiple-choice online self-assessment test was established, which the students could take repeatedly and at any time. The incoming test results were monitored continuously, and the students' preparatory training evaluated, based on factors such as the achieved test scores and the number of trials a student took of the test. Students encountering difficulties were invited to join assisted MATLAB trainings. This ensured that each participant had the necessary basic MATLAB knowledge when starting the project.

In the eight-month period before the project started, the development of the practical exercises and the planning of the project management was done by a core team of eight supervisors, who are members of five different institutes of the department. The core team determined the resources required, developed a MATLAB software environment to control the robots remotely from the host computer, created the laboratory exercises and demonstrations, and trained the other supervisors with respect to the defined learning targets.

In this project, NXT robots can be controlled directly via MATLAB from the host computers, instead of transferring compiled programs to the embedded NXT ARM microcontroller. Thus, applications are not limited by the NXT computing

environment, and users are able to employ complex MATLAB features like 3D-plots, graphical user interfaces (GUIs), regression analysis, filter operations, or interpolation algorithms performed on the host computer. Remote control and, thus, mobility for robots are provided via the wireless Bluetooth communication interface of the LEGO Mindstorms NXT firmware. Based on this communication protocol, the core team developed a new MATLAB toolbox, called “RWTH—Mindstorms NXT Toolbox,” which provides an interface between MATLAB and NXT robots. In periodic core team meetings, the software design was refined, and the contents of the laboratory exercises were discussed. The core team developed, documented, and finally circulated the exercises to the other supervisors of the department.

IV. RWTH—MINDSTORMS NXT TOOLBOX

Although there is a wide choice of commercial and free Mindstorms software for different programming languages [32], only two implementations use the MATLAB environment. The first software package [38], developed by T. Chikamasa, is based on control loops in a MATLAB and Simulink environment. It provides a simulation mode and an embedded coder, which does not allow the program code to be debugged step-wise. Also, it focuses on advanced control theory and requires an initial familiarity with Simulink, which can hardly be expected of freshman students. G. Gutt’s software package [39] uses the Bluetooth communication between an NXT robot and computer as a remote control. It does, though, require additional software to address the Bluetooth connection via MATLAB. Since the received Bluetooth packets are always buffered in files, this implementation is not suited for a direct and intuitive computer–robot communication.

Therefore, the novel RWTH—Mindstorms NXT Toolbox for MATLAB was developed, which provides a direct communication between NXT robots and MATLAB without using additional software. To this end, the core team implemented the LEGO Mindstorms Bluetooth communication protocol [40] in MATLAB, which allows users to control the NXT’s motors, read its sensors (touch, sound, light, and ultrasonic), or activate other system features like checking the battery level or playing a tone. These direct commands, specified by the protocol, represent one command layer of the toolbox functions structure. A global overview of the communication between MATLAB and NXT robot using the multilayered toolbox structure, which provides low- as well as high-level functions, is shown in Fig. 3.

Recently, MathWorks published free code to control LEGO Mindstorms NXT robots remotely [41]. This code might also be applicable to the project tasks, but it provides less functionality than the new toolbox.

In addition, the RWTH—Mindstorms NXT Toolbox provides high-level functions and control functions to support an intuitive and comfortable interface. The usability of the programming interface is essential for the success of a beginners’ course [15]. Therefore, low-level functions and direct commands are encapsulated, and a user interface in the style of other MATLAB toolboxes is given. Fig. 4 shows two short MATLAB examples for reading different sensor values using high-level functions. The toolbox content also includes detailed documentations, first-step

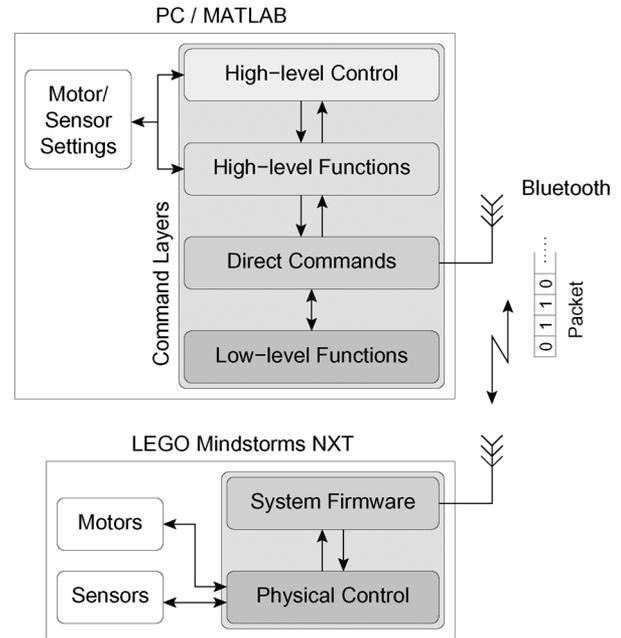


Fig. 3. Global overview of the communication between MATLAB and NXT hardware. The toolbox structure provides four command layers. The MATLAB user can apply motor and sensor settings and whole control loops via high-level functions. These configurations are then passed to the direct command layer, which creates a valid binary Bluetooth packet using low-level functions. In the next step, the packet is transmitted via the wireless Bluetooth channel to the NXT. The NXT system firmware decodes the message and triggers the physical control of the NXT motors and sensors. For hardware reading, the transmission chain is applied in reverse order.

```
>> OpenUltrasonic(SENSOR_1);
>> distance = GetUltrasonic(SENSOR_1)
distance = 36
>>

>> OpenSound(SENSOR_2, 'dB');
>> for i = 1:100
>>     sound = GetSound(SENSOR_2);
>>     if sound > threshold
>>         NXT_PlayTone(400, 500); % (400Hz, 500ms)
>>     end
>> end
```

Fig. 4. MATLAB program examples using high-level functions for reading NXT sensors. The upper example returns the current distance value in centimeters measured by the ultrasonic sensor, which is connected to port *SENSOR_1*. The lower example initializes the sound sensor at the second port and reads 100 samples in decibels. As soon as one sample exceeds the given threshold, a tone at 400 Hz is played.

instructions, and program examples. Thus, an individual’s own Mindstorms control programs can be easily implemented, even for MATLAB beginners.

The possibility of controlling the robots remotely reduces the development of autonomous robots to three main steps: building a robot, developing a MATLAB program, and executing the program on the host computer. An intermediate

Length, LSB	Length, MSB	Command Type	Command	Byte 5	Byte 6	...
-------------	-------------	--------------	---------	--------	--------	-----

Fig. 5. Structure of a valid Bluetooth packet, defined by the Bluetooth communication protocol [40]. The first two bytes determine the total length of the packet. The third byte specifies the command type, which can be a direct or a system command with or without acknowledgment. The next byte defines the command. The instruction $0x03$, for instance, corresponds to *PLAYTONE*. The last bytes of the packet carry the payload, such as the values of command parameters.

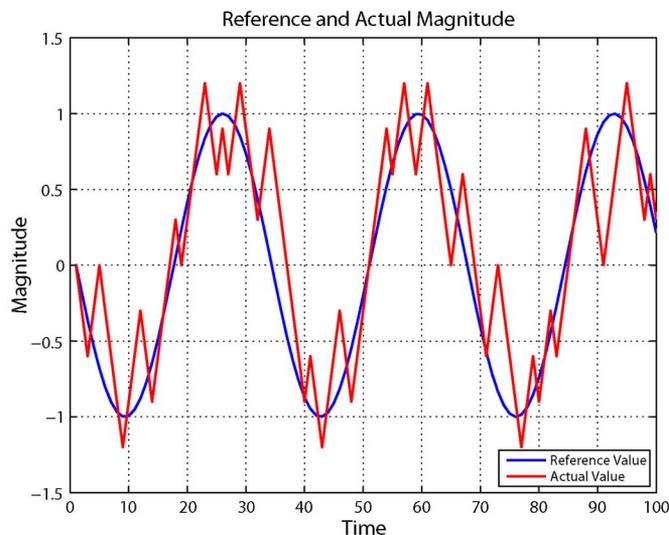
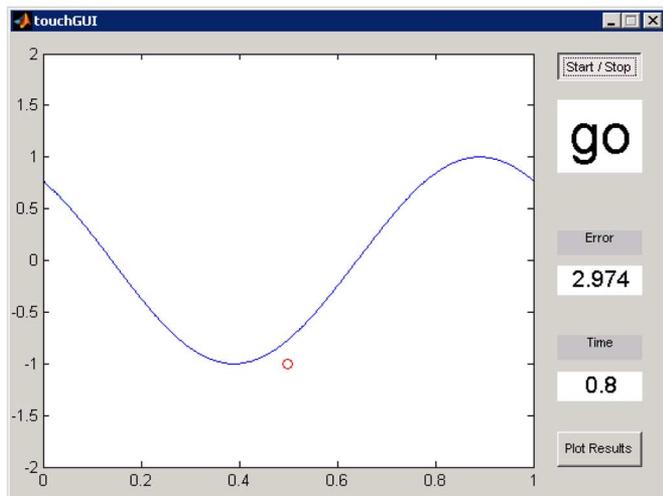


Fig. 6. Exercise 2: This GUI application emulates a moving sine wave. In the middle of the display, a cursor is plotted. The vertical upward or downward movement of the cursor is controlled by time intervals generated by manually pressing and releasing the NXT touch sensor. Over a 20-s time period, the user tries to track the sine wave with the cursor. (left) During this process, the current cursor position is calculated and displayed together with the moving sine wave on the right-hand side of the GUI. (right) After expiry of the 20-s interval, the reference sinusoid and the tracking result are displayed over time.

step, in which the program is compiled and transferred to the embedded NXT processor, is not required. Instead, a debugging and monitoring mode is supported, which allows the students to validate their program code during execution. The one relevant drawback of this design is caused by properties of the wireless Bluetooth channel. In general, the data rate of Bluetooth is limited to 1–3 Mb/s, and data errors and packet loss can occur randomly during data transmission. However, a bottleneck originates from the Bluecore chip of the NXT hardware, which needs a 30-ms latency to switch from receive to transmit mode [40]. This limits closed-loop control in real-time applications requiring a short response time via the NXT Bluetooth communication protocol.

The RWTH—Mindstorms NXT Toolbox for MATLAB supports Windows and Linux platforms and is published as a free, open-source product. It is subject to the GNU General Public License (GPL) [42] and can be downloaded from [43]. The establishment of an open-source software project seeks to induce others to participate in the project development and to found international interacting communities [44]. Other universities and schools are encouraged to introduce the toolbox into their own practical robotics courses. One example is the lecture “Introduction to Control Systems” at the School of Information Technology and Engineering, University of Ottawa, Ottawa, Canada [45].

The interface between NXT robots and MATLAB uses the host computer system for all control computations. Therefore, more complex features (like visualizations, optimization algorithms, profiler and debug information, and additional toolboxes) or libraries (such as for fuzzy logic, neural networks,

image processing, or databases) can be easily implemented and used for advanced robot control, going beyond the described educational purposes.

V. PRACTICAL PROJECT ACTIVITIES

A. Basic Exercises

In the first experiment of the laboratory exercises (Fig. 2), the students familiarize themselves with the NXT and its hardware configurations. Guided by the instruction manual prepared by the core team, they construct a driving robot and use the on-brick programming of the NXT firmware menu to make the robot follow a black line on a white sheet of paper. After only one hour’s effort, the students are able to have their first robots drive across the laboratory floor. This gives them an immediate sense of achievement and both motivation and confidence for the following exercises.

The second exercise introduces the Bluetooth communication between the robot and the host computer using MATLAB. First insights into the LEGO communication protocol with its structure, as illustrated in Fig. 5, are acquired. After establishing the wireless connection via MATLAB using high-level functions of the RWTH—Mindstorms NXT Toolbox, the students generate and interpret single Bluetooth packets by varying the arguments of single commands, like the tone frequency or duration time of `NXT.PlayTone`. Next, they implement a MATLAB function to read the touch sensor. Using a prepared GUI application (Fig. 6), a vertically moving cursor has to be controlled manually with the touch sensor such that it tracks a moving sine wave as closely as possible. Whenever the touch sensor is pressed, the cursor moves upward; otherwise, it moves downward. This exercise

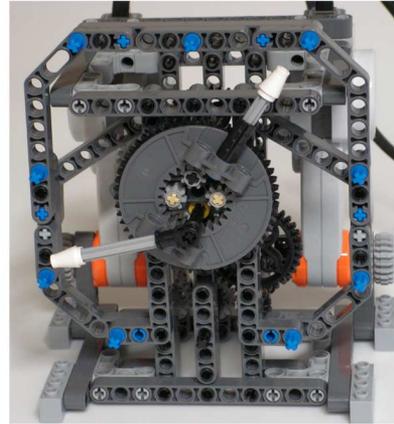
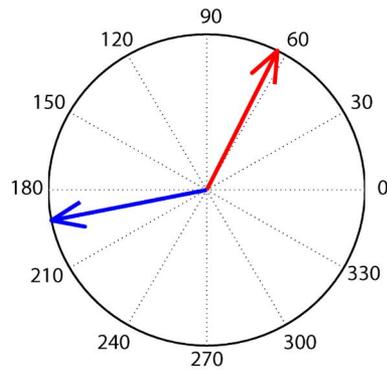


Fig. 7. Exercise 4: Calculation and visualization of complex numbers and their arguments. (left) Using the MATLAB compass plot, the arguments of two complex numbers are displayed on the host computer. (right) Corresponding to this argument plot, the same pointer positions are mechanically generated using a LEGO Mindstorms machine with intricate gear transmission ratios.

provides a simple example of sigma-delta modulation. Both the original sine wave and the tracking results are displayed in a MATLAB plot, as shown in Fig. 6. Also, the summed squared error is shown, thus allowing the students to compete with each other.

The third exercise focuses on programming structures like loops, if-conditions, and arrays. Here, the students implement a traffic light control system using lamps and the sound sensor. Depending on the number of detected handclaps, the lamps are switched on or off. The values of the sound sensor are continuously read and filtered by applying a moving sum to the data stream, thus providing for each time instance a value of how many claps occurred during the last 15 samples. At the time the students performed this exercise, the corresponding LSI-filter and its impulse response had already been treated in the lecture “Mathematical Methods of Electrical Engineering” (cf. [24, pp. 102, 145]). The different states of the traffic lights can then be determined by thresholding the filtered data stream.

The NXT servo motors are addressed in the fourth experiment. The students take measurements of the internal rotation sensor and work with different gear transmission ratios. Additionally, a machine to visualize complex numbers and vector arithmetics is built and programmed. In this subtask, two complex numbers and a mathematical operation read from a user input dialog are visualized with a MATLAB compass plot as well as with mechanical pointers of the LEGO Mindstorms machine, as shown in Fig. 7. As shown in the following, the direct relation between mathematics, MATLAB implementation, and robot control commands is here especially evident:

I Mathematics:

$$a = 4 + 8i = 4\sqrt{5}e^{j63.4349^\circ}$$

$$b = -5 - i = \sqrt{26}e^{-j168.6901^\circ}$$

II MATLAB:

```
>> a = 4 + 8i; b = -5 - 1i;
>> angles = angle([a,b]) / pi * 180
angles =
    63.4349 - 168.6901
>>
```

III NXT Control Commands:

```
>> NXT_angles = angles.*gear_ratios;
>> SetMotor(MOTOR_A); %(set motor A active)
>> SetPower(power); %(set motor power/speed)
>> SetAngleLimit(NXT_angles(1)); %(set angle)
>> SendMotorSettings; %(start motor rotation)
```

The fifth experiment focuses on light sensor measurements, programming with timers, and MATLAB GUI design. First, the students are introduced to the color sensitivity of the sensor compared to the human eye, as illustrated in Fig. 8 (left). Against the background of this sensor characteristic, the students take light measurements of differently colored objects using timer objects in MATLAB, which allow an automatic and periodic sensor reading. Furthermore, a GUI, including buttons and displays, has to be developed to control the timers. Fig. 8 (right) shows a final GUI implementation and the resulting plot of color measurements. Using the evaluated intensities, thresholds are determined for a robust classification of differently colored objects.

Finally, in the last mandatory exercise the students develop an explorer robot that scans its local environment using the ultrasonic sensor and drives autonomously through an open gate of an enclosure, as illustrated in Fig. 9. To complete this exercise, the students program the robot to make a 360° distance profile of the local environment and process the sensor data with basic filter operations like edge detection (calculated by moving differences, also already treated in the accompanying lecture) or median filtering.

B. Individual Tasks

In the second part of the laboratory, the students focus on more complex practical challenges. At this point, being in teams of four, they have more flexibility and are encouraged to develop their own ideas into robot systems. Alternatively, those who wish may select from a choice of predefined tasks. These include a parcours robot that enters, maps, and exits a maze; a 3-D robot arm that grabs and sorts colored balls (Fig. 10); and a 2-D scanner to scan poster-sized grayscale images shown in Fig. 11. To ensure comparability, all individual robot systems

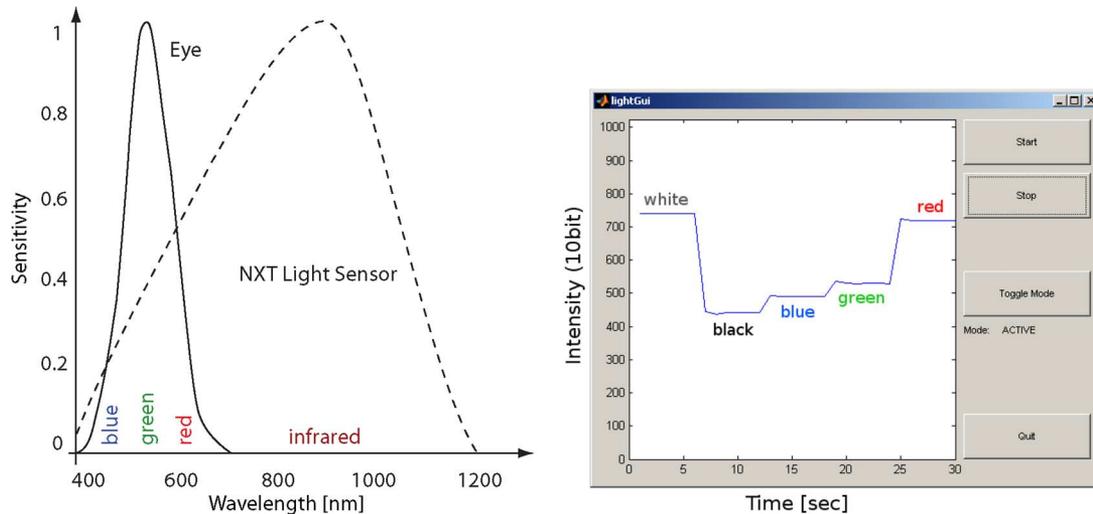


Fig. 8. Exercise 5: (left) Plot of the relative sensitivity of the NXT light sensor over wavelength, compared to the human eye. MATLAB GUI controls the reading of the light sensor triggered by timers. (right) In a 30-s interval, the intensity of white, black, blue, green, and red objects is measured with a 10-bit data resolution.

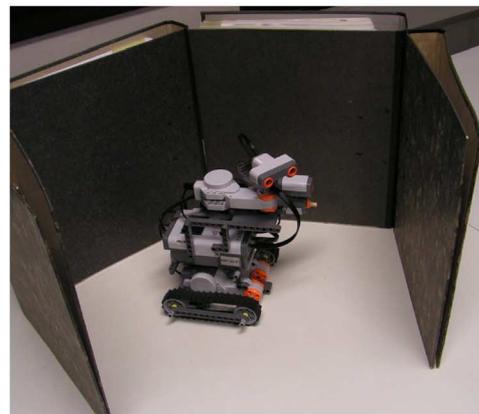
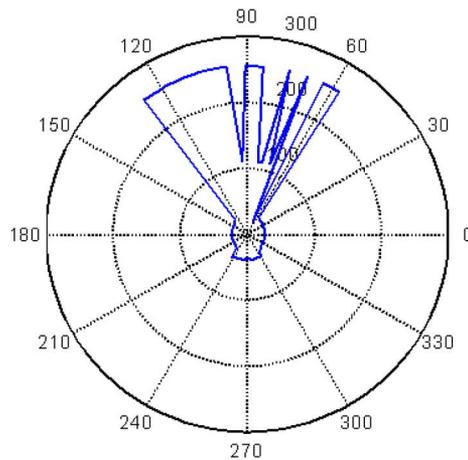


Fig. 9. Exercise 6: (left) The explorer robot scans the local environment and plots a 360° distance profile. (right) To find the open gate of the enclosure, the measured sensor values are processed with basic filter operations to eliminate outliers and to provide a robust estimation of the final direction.

are required to include data processing and programming with MATLAB, such as data visualization, mapping, GUI interaction, and graphical display.

For the 2-D scanner system, the students are instructed to construct a two-joint robot arm and to develop a time-efficient scanning algorithm. Here, the current position of the light sensor has to be calculated from the angle position of the two joints, as shown in Fig. 11. Problems such as transformation from polar to Cartesian coordinates, sampling, image interpolation, and visualization are addressed in subtasks. After the students have successfully plotted the sensor data values in Cartesian coordinates, they are introduced to mean value interpolation to fill the gaps of the sampled image grid. Without an in-depth treatment of sampling and image processing methods, the interpolation is described as the evaluation of the local neighborhood of a pixel via a 2-D LSI filter mask, as shown in Fig. 12. Although MATLAB provides advanced filter functions, for-loop implementations can also be used to pass this exercise. Fig. 13 illustrates the two image reconstruction steps. It shows the raw image after scanning and the interpolation result. Despite the hardware limitations, the interpolated image is of reasonable quality.

C. Presentation

On the last project day, the student teams present their results and individual robot constructions to the other participants and the supervisors. Each team is allocated a 15-min time interval for the presentation. The teams are requested to give live demonstrations of the integrated features and behavior of their robots and to outline their algorithms and MATLAB implementations. As presentation tools, the students use common software like OpenOffice.org or Microsoft PowerPoint.

VI. EVALUATION AND RESULTS

To earn the credit points for the project, the students have to pass all basic exercises, carry out at least one individual exercise, and present their project work as described above. After completion of an exercise, each student group delivers its results and MATLAB implementations to a supervisor. If an implementation solved the exercise but was computationally inefficient, individual suggestions to improve the solutions were given, and

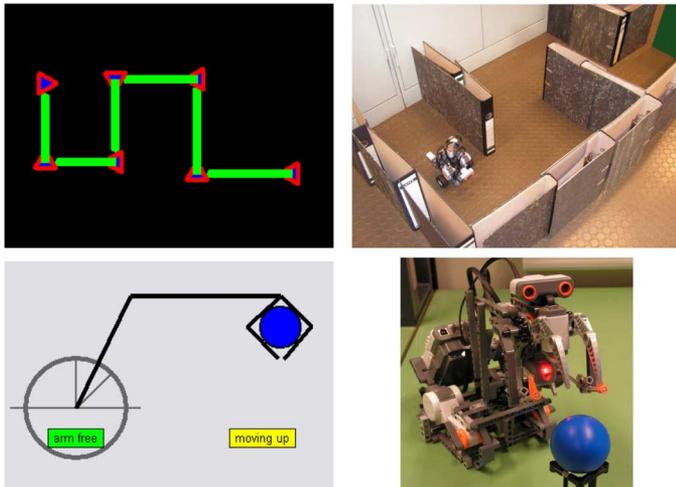


Fig. 10. (top right) The parcours robot drives autonomously through a maze, scans its environment with ultrasonic distance measurements, and (top left) maps the route online in a MATLAB plot. (bottom right) The 3-D robot arm grabs and sorts colored balls and (bottom left) sends its current processing state and position to a MATLAB monitor display.



Fig. 11. The 2-D scanner scans a poster image using the NXT light sensor. The position of the sensor is determined by a two-joint robot arm controlled by servo motors. The angle position of the two joints can be described by the angles α and β .

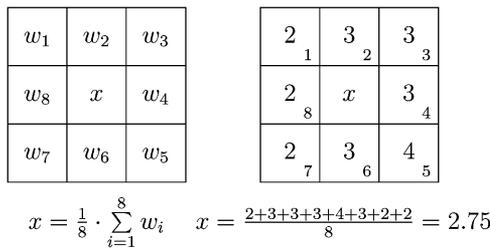


Fig. 12. Mean value interpolation. (left) Illustration of the local eight-neighborhood pixels $w_1 \dots w_8$ (2-D filter mask) of the current image pixel position x . (right) Example of interpolation by the local average.

the sample solution as prepared by the core team was eventually made available to the students to guide them toward structured programming methods. On average, one tutor supervised five students. If students were absent on project days, they were given the opportunity to complete the missed exercises on an additional project day. In total, 93% of the 309 participants passed the project. This rate shows that almost all students met or even exceeded the requirements to pass each exercise and achieved the project goals.

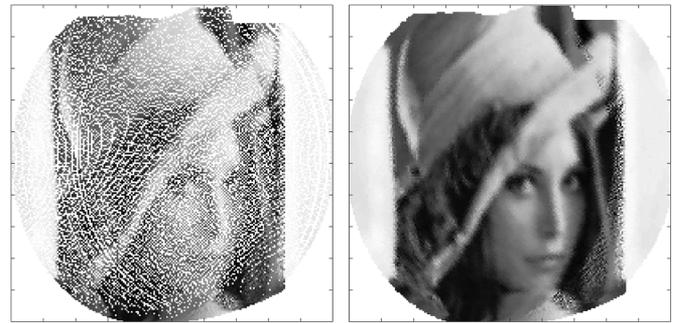


Fig. 13. 2-D scan of a poster-size image, recorded by the NXT light sensor. After scanning and a coordinate transform from polar to Cartesian coordinates, the sampled image shown on the left results. Using a moving average filter for interpolation fills the gaps between the samples, as shown on the right.

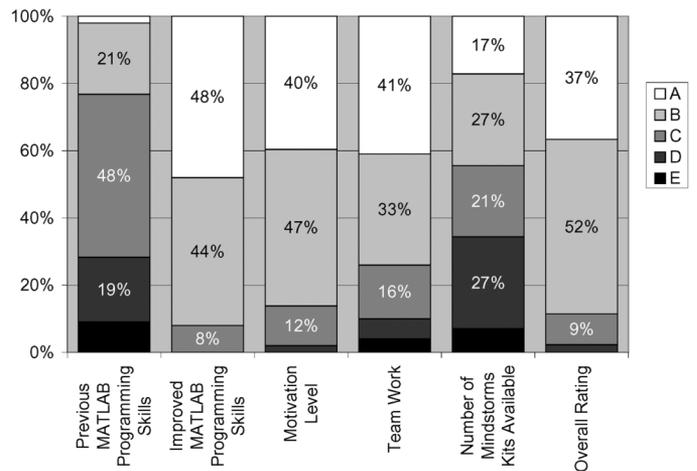


Fig. 14. Evaluation results of the project based on an anonymous student online evaluation (A: excellent, B: good, C: average, D: below average, E: inadequate).

After the laboratory had ended, an anonymous and voluntary online evaluation was carried out. There were 131 students who participated in the survey, answering up to 38 questions about the general project concept, specific exercises, and personally achieved improvements. The summarized evaluation results in Fig. 14 show that the laboratory course also achieved the goal of introducing MATLAB as a software to solve real-world problems from the students' point of view. 48% of the students rated their MATLAB programming skills as "average" at the beginning of the laboratory, whereas 48% rated the improvement of their programming skills as "excellent" and 44% as "good" after the project. Even though these results are based on the students' own subjective self-assessments, they agree well with the skills objectively required to pass the project, such as the ability to develop and program one's own individual robot via MATLAB, based on mathematical methods and principles. Moreover, these observations are consistent with experiences in other project-based courses using robot kits, which report an improvement of the students' programming skills [15], [33], [34], [46].

When asked to "identify positive aspects of the laboratory," the students replied with individual statements such as "confrontation with problems which are not treated in theoretically

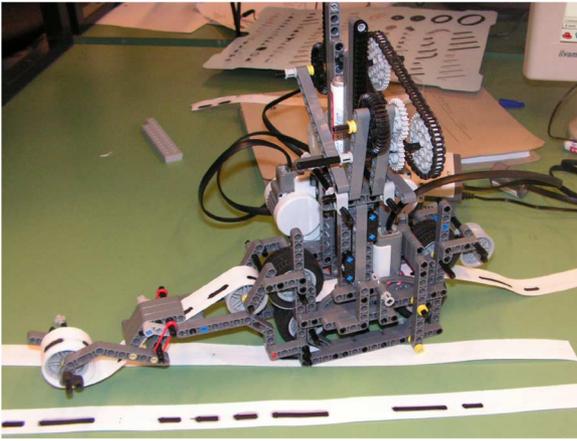


Fig. 15. Morse encoder and decoder, invented and developed by a student team. It encodes a text, provided by the user through a MATLAB GUI, in Morse code and writes the symbols on a moving paper tape using a black marker. Conversely, when provided with Morse code written on the paper tape, the machine reads the symbols with the NXT light sensor and decodes the coded message into plain text.

oriented lectures,” “theoretical foundations could be linked to practical problems, which led to a higher level of understanding,” and “practice-oriented application with MATLAB.” These are another indicator that the project objectives were achieved. The positive student feedback goes along with a high level of motivation: 40% of the students rated their motivation as “excellent,” and another 47% as “good.” Fig. 14 also shows the rating of the teamwork within the student groups. 74% of the students evaluated the collaboration with their fellow students as “excellent” or “good.” Altogether, the overall rating of the project was graded as “excellent” or “good” by 89% of all feedback responses.

The students’ creativity and the variety of individual application ideas, robot constructions, and inventions exceeded the expectations. The evaluation showed that 45% of the survey participants defined and implemented their own individual robot systems. Examples of these include a 3-D environment scanner with target recognition, a bottle finder with fill-level-scanner, a vehicle with roadside recognition, a bowling robot, an autonomous parking vehicle, a sorting machine, a barcode disk scanner, and a Morse encoder and decoder. As an example, the Morse encoder and decoder is shown in Fig. 15.

While many students argued that working with the hardware limitations was challenging, some participants criticized the randomly occurring major Bluetooth latencies and the complex motor control, which made the implementation of their ideas more complicated. Also, as one robot kit had to be shared between two student pairs during the basic exercises, many participants would have wished for more robot kits to be available (see the scoring related to the “Number of Mindstorms Kits Available” in Fig. 14). Still, 87% of the students said they would join the practical course again and would like to see LEGO Mindstorms robot kits also used in other projects and free-time workshops.

The MATLAB online test was rated as “excellent” or “good” by 76% of the students, showing that the self-assessment test is well suited to prepare students for the project. The MATLAB Primer and the exercise documentation provided were used by 95% of the survey participants for preparation.

Other questions of the online evaluation addressed topics related to each individual exercise, like the time needed to complete it, quality of the exercise documentation, and the level of difficulty. Feedback to these questions revealed no major shortcomings and was the basis for minor revisions for the next project term.

Overall, the evaluation results clearly show that the freshman students were highly motivated during the entire eight project days. The laboratory concept also helped the students to acquire social skills, like working in a team, managing the assigned tasks within a given time frame, solving occasional conflicts, presenting their results, and cooperating and communicating with peers to attain the educational goals.

The observations made by the supervisors during the project were also collected. They reported observing active group dynamics, enthusiasm, competition between teams, and even students spending additional time on refining their robot algorithms after the official hours. Furthermore, the supervisors observed peer learning processes within and across the teams when the students shared laboratory resources and engaged in debates. Finally, it should also be mentioned that during the examination term following the first semester, the average grading and the number of students who passed were highest for the subjects with contents related to the project, i.e., “Basics of Computer Science I” (introduction to programming techniques, data structures, and algorithms) and “Mathematical Methods of Electrical Engineering.” (All examinations were written; other subjects with examinations in the same term included Circuit Theory I, Mathematics, and Physics.) Although not an absolute proof, this observation is another indicator that the project succeeds in imparting programming and other skills to the students.

VII. PERSPECTIVES

Based on these first experiences and the evaluation results, the number of Mindstorms NXT education sets will be doubled for the next laboratory course to accelerate the execution time of the basic mandatory exercises and to provide additional time for the individual tasks. The RWTH—Mindstorms NXT Toolbox for MATLAB will be adapted to the USB interface to avoid major packet loss and enhance the computer–robot communication for stationary robots. Also, the syntax of the high-level functions of the toolbox will be extended and simplified, especially where student feedback revealed room for improvements. As mentioned above, minor adjustments of specific exercises will be made based on the students’ and the instructors’ feedback.

Due to the great demand, a voluntary workshop for students who wish to continue their projects in their spare time has been established. Finally, this laboratory might also be integrated into the curriculum of Industrial Engineering.

VIII. CONCLUSION

In this paper, a new freshman introduction course into practical engineering is presented. It is organized as a full-time block course over eight days, with a threefold learning concept to strengthen mathematical basics, foster MATLAB programming skills, and introduce students to real engineering problems. The project structure including mandatory and individual exercises is described, where the objective is to induce the students to transfer mathematical methods to robot applications via algorithm development, MATLAB programming, and constructing robots. To provide a more intuitive and transparent computer–robot communication via MATLAB, a new toolbox was developed and is made publicly available as a free open-source project.

The percentage of students who passed the project, the unexpectedly wide variety of robot systems created during the second project phase, and the evaluation results of both students and supervisors show that the project augmented student motivation and helped the students to develop their programming skills and acquire soft skills. The mathematical tasks, mechatronical issues, and inherent hardware limitations were challenging for most of the participants and led to intensive group dynamics, teamwork, competition, engagement, and remarkable creativity. Overall, the freshman introduction course met the practical engineering and MATLAB programming learning targets.

Stimulated by these results, the extension of the project concept to other practical courses in the curriculum, comprising also interdisciplinary student groups, is currently under consideration.

IX. PROJECT WEB PAGE

Further images, videos, and descriptions of robot constructions created by different students groups can be found at the project Web page: <http://www.lfb.rwth-aachen.de/mindstorms>.

The free download and contact information for the developer community of the open-source software project “RWTH—Mindstorms NXT Toolbox for MATLAB” is available at <http://www.mindstorms.rwth-aachen.de>.

REFERENCES

- [1] S. Director, P. Khosla, R. Rohrer, and R. Rutenbar, “Reengineering the curriculum: Design and analysis of a new undergraduate electrical and computer engineering degree at carnegie mellon university,” *Proc. IEEE*, vol. 83, no. 9, pp. 1246–1269, Sep. 1995.
- [2] R. Saint-Nom and D. Jacoby, “Building the first steps into SP Research,” in *Proc. IEEE ICASSP*, 2005, vol. 5, pp. 545–548.
- [3] M. Vallim, J.-M. Farines, and J. Cury, “Practicing Engineering in a Freshman Introductory Course,” *IEEE Trans. Educ.*, vol. 49, no. 1, pp. 74–79, Feb. 2006.
- [4] D. Gabauer, T. Bayse, J. Terpenney, and R. Goff, “Improving undergraduate engineering design instruction through lessons learned mentoring first LEGO league,” presented at the Annu. Conf. Amer. Soc. Eng. Educ., Jun 2007.
- [5] R. Pop-Iliev and S. B. Nokleby, “Concurrent approach to teaching concurrent design engineering,” presented at the 2nd Int. Conf. Design Educ., Innovation, and Practice, Jul. 2005.
- [6] G. W. Georgi, L. M. Folan, and D. R. Doucette, “Introducing design throughout the curriculum,” presented at the Annu. Conf. Amer. Soc. Eng. Educ., Jun. 2002.
- [7] J. Vicente, B. Garcia, A. Mendez, I. Ruiz, and O. Lage, “EasySP: The easiest form to learn signal processing interactively,” in *Proc. IEEE ICASSP*, Apr. 2007, vol. 3, pp. 713–716.
- [8] E. Delp and Y.-H. Lu, “The use of undergraduate project courses for teaching image and signal processing techniques at Purdue University,” in *Proc. 12th Signal Process. Educ. Workshop*, Sep. 2006, pp. 281–284.
- [9] R. Saint-Nom and D. Jacoby, “Sigma-delta converters as a sp teaching tool,” in *Proc. IEEE ICASSP*, May 2006, vol. 2, pp. II-II.
- [10] F. Klassner and S. Anderson, “LEGO MindStorms: Not Just for K-12 Anymore,” *IEEE Robot. Autom. Mag.*, vol. 10, no. 2, pp. 12–18, Jun. 2003.
- [11] N. Azlan, F. Zainudin, H. Yusuf, S. Toha, S. Yusoff, and N. Osman, “Fuzzy logic controlled miniature LEGO robot for undergraduate training system,” in *Proc. 2nd IEEE ICIEA*, May 2007, pp. 2184–2188.
- [12] D. Ye, I. Brutman, G. Georgi, and L. Folan, “Freshman project: Autonomous underwater vehicle (AUV),” presented at the Annu. Conf. Amer. Soc. Eng. Educ., 2007.
- [13] F. Michaud, “Engineering education and the design of intelligent mobile robots for real use,” *Int. J. Intell. Autom. and Soft Comput.*, vol. 13, no. 1, pp. 19–28, 2007.
- [14] M. Neilsen, “Research Experiences in Robotics,” in *Proc. Int. Conf. Soc. Inf. Technol. and Teacher Educ.*, 2006, no. 1, pp. 3914–3919.
- [15] D. Cliburn, “Experiences with the LEGO mindstorms throughout the undergraduate computer science curriculum,” in *Proc. 36th FIE Conf.*, Oct. 2006, pp. 1–6.
- [16] R. Maher, J. Becker, T. Sharpe, J. Peterson, and B. Towle, “Development and implementation of a robot-based freshman engineering course,” in *Proc. Annu. Conf. Amer. Soc. Eng. Educ.*, Jun. 2005.
- [17] M. Christensen, S. Douglas, S. Wood, C. Kitts, and T. Mahler, “The infinity project brings DSP brains to robots in the classroom,” in *Proc. 3rd IEEE Signal Process. Educ. Workshop*, Aug. 2004, pp. 88–91.
- [18] C. Pomalaza-Raez and B. H. Groff, “Retention 101: Where robots go . . . students follow,” *J. Eng. Educ.*, vol. 92, no. 1, pp. 1–6, Jan. 2003.
- [19] P. Lau, S. McNamara, C. Rogers, and M. Portsmore, “LEGO robotics in engineering,” in *Proc. Annu. Conf. Amer. Soc. Eng. Educ.*, Jun. 2001.
- [20] T. Hissey, “Education and careers 2000. Enhanced skills for engineers,” *Proc. IEEE*, vol. 88, no. 8, pp. 1367–1370, Aug. 2000.
- [21] P. E. Goodmann, “Teaching ECET students to be self-directed learners with first-year projects and amateur radio,” *J. Eng. Technol.*, Oct. 2007.
- [22] R. J. Roselli and S. P. Brophy, “Effectiveness of challenge-based instruction in biomechanics,” *J. Eng. Educ.*, vol. 93, no. 4, pp. 311–324, Oct. 2006.
- [23] M. Yoder, J. McClellan, and R. Schafer, “Crystal radios or DSP First?,” in *Proc. 28th FIE Conf.*, 1998, vol. 2, pp. 695–699.
- [24] J. McClellan, R. Schafer, and M. Yoder, *Signal Processing First*. Englewood Cliffs, NJ: Prentice-Hall, Nov. 2002.
- [25] J. McClellan, R. Schafer, and M. Yoder, “Experiences in teaching DSP first in the ECE curriculum,” in *Proc. 27th FIE Conf.*, Nov. 1997, vol. 2, pp. 891–895.
- [26] D. Anderson, J. McClellan, R. Schafer, J. Schodorf, and M. Yoder, “DSP first—A first course in ECE,” in *Proc. 13th Asilomar Conf. Signals, Syst., and Comput.*, 1996, vol. 1, pp. 226–230.
- [27] MATLAB, The MathWorks [Online]. Available: <http://www.mathworks.com>
- [28] M. Yoder and B. Black, “Teaching DSP first with LabVIEW,” in *Proc. 4th IEEE Signal Process. Educ. Workshop*, Sep. 2006, pp. 278–280.
- [29] “Bologna Declaration,” European Ministers of Education, Jun. 1999 [Online]. Available: <http://www.ond.vlaanderen.be/hogeronderwijs/bologna/documents/>
- [30] LEGO MINDSTORMS [Online]. Available: <http://www.mindstorms.com>
- [31] V. Dagdilelis, M. Sartazemi, and K. Kagani, “Teaching (with) robots in secondary schools: Some new and not-so-new pedagogical problems,” in *Proc. 5th IEEE ICALT*, Jul. 2005, pp. 757–761.
- [32] H. Patterson-McNeill and C. L. Binker, “Resources for using LEGO Mindstorms,” *J. Comput. Sci. Coll.*, vol. 16, no. 3, pp. 48–55, 2001.
- [33] A. Williams, “The qualitative impact of using LEGO MINDSTORMS Robots to Teach Computer Engineering,” *IEEE Trans. Educ.*, vol. 46, no. 1, p. 206, Feb. 2003.

- [34] M. I. G. Mota, "Work in progress—Using lego mindstorms and robolab as a mean to lowering dropout and failure rate in programming course," in *Proc. 37th FIE Conf.*, Oct. 2007, pp. F4A1–F4A2.
- [35] S. Sharad, "Introducing embedded design concepts to freshmen and sophomore engineering students with LEGO MINDSTORMS NXT," in *Proc. IEEE Int. Conf. Microelectron. Syst. Educ.*, Jun. 2007, pp. 119–120.
- [36] "LEGO education," [Online]. Available: <http://www.legoeducation.com>
- [37] C. McGoldrick and M. Huggard, "Peer learning with lego mindstorms," in *Proc. 34th FIE Conf.*, Oct. 2004, pp. S2F24–S2F29.
- [38] T. Chikamasa, "Embedded coder robot NXT demo," accessed Feb. 18, 2009 [Online]. Available: <http://www.mathworks.com/matlabcentral/fileexchange/>
- [39] G. Gutt, "Controlling NXT from MathWorks MATLAB," 2006 [Online]. Available: <http://nxtasy.org/2006/11/28/controlling-nxt-from-mathworks-matlab/>, accessed on Feb. 18, 2009.
- [40] LEGO MINDSTORM Bluetooth Developer Kit 2007 [Online]. Available: <http://mindstorms.lego.com/Overview/NXTreme.aspx>
- [41] "Programming LEGO MINDSTORMS NXT robotics with MATLAB and Simulink," The MathWorks, 2008 [Online]. Available: <http://www.mathworks.com/programs/lego/>
- [42] "GNU general public license," 2007 [Online]. Available: <http://www.gnu.org/licenses/licenses.html>
- [43] RWTH—Mindstorms NXT Toolbox for MATLAB, RWTH Aachen University, Aachen, Germany, 2008 [Online]. Available: <http://www.mindstorms.rwth-aachen.de>
- [44] A. Hars and S. Ou, "Working for free?—Motivations of participating in open source projects," in *Proc. 34th Int. Conf. Syst. Sci.*, Jan. 2001, pp. 1–9.
- [45] "Introduction of control systems," School of Information Technology and Engineering, University of Ottawa, Ottawa, Canada, Summer Term 2008, accessed on Feb. 18, 2009 [Online]. Available: <http://www.site.uottawa.ca/~rhabash/ELG315008.html>
- [46] M. Garcia and H. Patterson-McNeill, "Learn how to develop software using the Toy LEGO mindstorms," in *Proc. 32nd FIE Conf.*, Nov. 2002, vol. 3, pp. 7–10.

Alexander Behrens (S'09) was born in Bückeberg, Germany in 1980. He received the Dipl.-Ing. degree in electrical engineering from the Leibniz University of Hannover, Hannover, Germany, in 2006.

After receiving the degree, he worked as a Research Assistant at the university's Institut für Informationsverarbeitung. Since 2007, he has been a Ph.D. candidate at the Institute of Imaging and Computer Vision, RWTH Aachen University, Aachen, Germany. His research interests are in medical image processing, signal processing, pattern recognition, and computer vision.

Linus Atorf was born in Düsseldorf, Germany in 1984. Since 2004, he has been studying physics at RWTH Aachen University, Aachen, Germany, and since 2007, he has been a Student Assistant at the university's Institute of Imaging and Computer Vision.

Robert Schwann was born in Dortmund, Germany. He received the Dipl.-Ing. degree from RWTH Aachen University, Aachen, Germany, in 1997.

Since then, he has been at the Institute of Electrical Engineering and Computer Systems, RWTH Aachen University. His field of work is the signal processing and image quality of medical ultrasound and OCT systems.

Bernd Neumann received the B.Eng. degree in electrical and electronics engineering from Coventry University, Coventry, U.K., in 2000; the Dipl.-Ing. (First Honors) degree in electrical engineering from the Aachen University of Applied Sciences, Aachen, Germany, in 2000; and the M.Sc. degree in computer engineering from RWTH Aachen University, Aachen, Germany, in 2002.

In 2002, he joined the Institute of Electrical Engineering and Computer Systems at RWTH Aachen University as a Research Assistant. Since 2008, he has been with Robert Bosch GmbH, Reutlingen, Germany, as an ASIC-designer in the field of automotive electronics.

Rainer Schnitzler was born in Mönchengladbach, Germany. He studied mathematics and computer science at RWTH Aachen University, Aachen, Germany.

Since 1987, he has worked as Software Developer, Computer Administrator, and Training Supervisor at the Institute of Electrical Engineering and Computer Systems, RWTH Aachen University.

Johannes Ballé was born in Dortmund, Germany in 1980. He received the Dipl.-Ing. degree in computer engineering from RWTH Aachen University, Aachen, Germany, in 2007.

Currently, he is pursuing the Ph.D. degree at the Institute of Communications Engineering (IENT), RWTH Aachen University, focusing on image processing, signal theory, and data compression.

Thomas Herold was born in 1977 in Daun, Germany. He received the Dipl.-Ing. degree in electrical engineering from RWTH Aachen University, Aachen, Germany, in April 2007.

Since then, he has been a Ph.D. candidate at the Institute of Electrical Machines, RWTH Aachen University. His main field of research is propulsion control and system identification.

Aulis Telle was born in Leverkusen, Germany. He received the Dipl.-Ing. degree in electrical engineering from RWTH Aachen University, Aachen, Germany, in 2004.

Since then, he has been a Ph.D. candidate at the Institute of Communication Systems and Data Processing, RWTH Aachen University. His research interests are in digital signal processing for medical applications and system identification in noisy environments.

Tobias G. Noll (SM'03) received the Ing. (grad.) degree in electrical engineering from the Fachhochschule Koblenz, Koblenz-Oberwerth, Germany, in 1974; the Dipl.-Ing. degree in electrical engineering from the Technical University of Munich, Munich, Germany, in 1982; and the Dr.-Ing. degree from the Ruhr-University of Bochum, Bochum, Germany, in 1989.

From 1974 to 1976, he was with the Max-Planck-Institute of Radio Astronomy, Bonn, Germany, being active in the development of microwave waveguide and antenna components. In 1976, he joined the Corporate Research and Development Department of Siemens, and in 1987, he headed a group of laboratories concerned with the design of algorithm specific integrated CMOS circuits for high-throughput digital signal processing. In 1992, he joined the Electrical Engineering Faculty of RWTH Aachen University, Aachen, Germany, where he is a Professor, holding the Chair of Electrical Engineering and Computer Systems. In addition to teaching, he is involved in research activities on VLSI architectural strategies for high-throughput digital signal processing, circuits concepts, and design methodologies with a focus on low-power CMOS and deep submicron issues, as well as on digital signal processing for medical electronics.

Kay Hameyer (SM'99) received the M.Sc. degree in electrical engineering from the University of Hannover, Hannover, Germany, and the Ph.D. degree from the University of Technology (TU) Berlin, Berlin, Germany.

After his university studies, he worked with the Robert Bosch GmbH in Stuttgart, Germany, as a design engineer for permanent magnet servo motors and board net components. In 1988, he became a member of the staff at the TU Berlin. From November to December 1992, he was a Visiting Professor at the COPPE Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil, teaching electrical machine design. In the frame of collaboration with the TU Berlin, he was a Visiting Professor at the Université de Batna, Batna, Algeria, in June 1993. Beginning in 1993, he was a Scientific Consultant working on several industrial projects. Currently, he is a Guest Professor at the University of Maribor, Maribor, Slovenia, and the Korean University of Technology and Education (KUTE), Cheonan, Korea. He is a Member of the Faculty of Electrical Engineering of the Technical University of Poznan, Poznan, Poland, and a Member of the Faculty of Electrical Engineering of the Technical University of Cluj-Napoca, Cluj-Napoca, Romania, which awarded him the academic title of a "Doctor honoris causa" in April 2004. Until February 2004, he was

a Full Professor for Numerical Field Computations and Electrical Machines with the Katholieke Universiteit Leuven, Leuven, Belgium. Currently, he is the Director of the “Institute of Electrical Machines” and holder of the Chair “Electromagnetic Energy Conversion” of RWTH Aachen University, Aachen, Germany (<http://www.iem.rwth-aachen.de/>) and Dean of the Faculty of Electrical Engineering and Information Technology. He is author of more than 150 journal publications, more than 250 international conference publications, and four books. His research interests are numerical field computation, the design of electrical machines, in particular permanent magnet excited machines, induction machines, and numerical optimisation strategies.

Prof. Hameyer is an Elected Member of the Board of the International Compumag Society, a Member of the German VDE, a Fellow of the IEE (IET), and a Member of the Executive Team of the IEE Professional Network “Electromagnetics.”

Til Aach (M’94–SM’02) received the diploma and doctoral degrees, both with honors, in electrical engineering from RWTH Aachen University, Aachen, Germany, in 1987 and 1993, respectively.

While working toward the doctoral degree, he was a Research Scientist with the Institute for Communications Engineering, RWTH Aachen University, being in charge of several projects in image analysis, 3D-television, and medical image processing. In 1984, he was an intern with Okuma Machinery Works Ltd., Nagoya, Japan. From 1993 to 1998, he was with Philips Research Labs, Aachen, Germany, where he was responsible for several projects in medical imaging, image processing, and analysis. In 1996, he was also an Independent Lecturer with the University of Magdeburg, Magdeburg, Germany. In 1998, he was appointed a Full Professor and Director of the Institute for Signal Processing, University of Luebeck, Luebeck, Germany. In 2004, he became Chairman of the Institute of Imaging and Computer Vision, RWTH Aachen University. He has authored or co-authored over 250 papers. He is a co-inventor on some 20 patents. His research interests are in medical and industrial image processing, signal processing, pattern recognition, and computer vision.

Prof. Aach has received several awards—among these, the award of the German “Informationstechnische Gesellschaft” (ITG/VDE) for a paper published in the IEEE TRANSACTIONS ON IMAGE PROCESSING in 1998. From 2002 to 2008, he was an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING. He was a Technical Program Co-Chair for the IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI) in 2000, 2002, 2004, and 2006. He is a member of the Bio-Imaging and Signal Processing Committee (BISP-TC) of the IEEE Signal Processing Society.