Control of an Embedded System via Internet

Gábor Sziebig, *Member, IEEE*, Béla Takarics, *Member, IEEE*, and Péter Korondi, *Member, IEEE*

***Abstract*—This paper presents our experience with a complete multimedia educational program of dc servo drives for distant learning. The program contains three parts: animation, simulation, and Internet-based measurement. The animation program helps to understand the operation of dc motors as well as its time- and frequency-domain equations, transfer functions, and the theoretical background necessary to design a controller for dc servo motors. The simulation model of the dc servo motor and the controller can be designed by the students based on the animation program. The students can also test their controllers through the Internet-based measurement, which is the most important part from an engineering point of view. Students can then perform various exercises such as programming the D/A and A/D cards in the embedded system and designing different types of controllers. First, a simple PI controller can be designed, but advanced students can also design more sophisticated controllers such as the sliding mode controller. After the measurements are executed, the students can download the measured data and compare them to the simulation results.**

***Index Terms*—DC motor, distant learning, sliding-mode control.**

I. Introduction

TWO e-learning education projects supported by the European Union provide the background of this paper. The contributors developing the projects were spread all over Europe and came from more than ten countries. The support was granted to the participants under the umbrella of the Leonardo da Vinci program (part of the European Commission's Lifelong Learning Programme).

One of the projects is called E-learning Distance Interactive Practical Education (EDIPE) [1]–[3]. It is the logical continuation of the second project called Interactive and Unified E-based Education and Training in Electrical Engineering (INETELE) which was completed earlier [4]–[7]. The animations can be downloaded free of charge.

Remote Laboratory Project (EDIPE): It is crucial for students to have some laboratory practice. The experiments give the students an insight into the real processes of the tested system and highly enhance their capability to apply theories learned in classes and from books. They can also see the influence of the second/higher order effect, as well as the effect of parasitics which are difficult to simulate perfectly.

However, it is rather expensive and time consuming to build experiments and keep them up to date. On the other hand, both the students and the universities can mutually benefit from a remotely accessible laboratory network available via the Internet. Each test in the laboratory network is worked out and kept up to date by different participating universities. The experiments, in general, can be conducted either online or based on recorded material (virtual experiment). It allows students to perform experiments safely and without guidance, and official working hours in the laboratory do not limit the users. The students can also experience the appearance of the measurement instrument, the electronic components, and many more factors such as layout. In general, the measurements should be designed as a project with educational philosophy. The experiments should be not only analysis oriented (to measure, to see, to evaluate, and to explain the results) but also synthesis oriented. They should involve design aspects.

The target of project EDIPE is to create more than 18 laboratories located in universities spread all over Europe. They will be accessible round the clock by authorized students. The laboratory tests can be run by them via Internet. The distant laboratory network multiplies the effectiveness of the hardware and software investments of the local staff for the benefit of all participants. The outputs will be teaching materials in electronic form incorporating description of the tests, guidelines, manuals, documentation, etc., in English. The topics of the laboratory tests cover a large range of the fundamentals and basics of power electronics, electric drives, and motion controls.

E-learning (INETELE): The targets and results in classical e-learning approaches, in general, did not contain the access to remote laboratories. They concentrated on the elaboration, delivery, and dissemination of knowledge, information similar to books, but they have been using electronic media, widening the border of communication tremendously, abolishing most of the limitations, and exploiting many new possibilities which were not at the disposal in the classical publishing technique. The e-learning elevated the communication of sophisticated scientific knowledge to higher dimensions and made the understanding and comprehension much easier. They can be used in classrooms, at home, and for distance learning either from local electronic storage or via Internet. They incorporate multimedia-rich pages with animations, color figures, and audio effects (speech and music). Other multimedia techniques like extensive usage of video clips, audio, or slide shows are also available. Advanced materials use interactivity, meaning the combination of text explaining the theory with interactive programs that

Manuscript received November 10, 2008; revised August 7, 2009; accepted December 23, 2009. Date of publication February 8, 2010; date of current version September 10, 2010. This work was supported in part by the National Science Research Fund (OTKA K62836) and in part by the Control Research Group of the Hungarian Academy of Science.

G. Sziebig is with the Department of Industrial Engineering, Narvik University College, 8514 Narvik, Norway (e-mail: gabor.sziebig@hin.no).

B. Takarics is with the Department of Mechatronics, Optics and Applied Informatics, Budapest University of Technology and Economics, 1521 Budapest, Hungary (e-mail: takarics@get.bme.hu).

P. Korondi is with the Computer and Automation Research Institute, 1111 Budapest, Hungary, and also with the Department of Mechatronics, Optics and Applied Informatics, Budapest University of Technology and Economics, 1521 Budapest, Hungary (e-mail: korondi@sztaki.hu).

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/TIE.2010.2041132

allow student to do little experiments via a simulator or solve some engineering problems.

The ways for keeping contacts between teacher and student and among the students are widely extended via e-mail, chat rooms, online tests, etc.

The program called INETELE incorporates and offers most of the features and properties just summarized, but the access to remote laboratories and experiments. The whole program is very ambitious, covering a large area of electrical engineering. We confine here the discussion of its very small section dealing with the control of power electronic and motion control developed by us. It will illustrate some of the main features of the program.

The organization of this paper is as follows. Section II summarizes the structure, the ways of presentation, and the interactive animations applied in the e-learning material by presenting and commenting a few characteristic pages. Section III describes the techniques and some of the conceptual issues in the development of Internet-based remote laboratory network. Section IV summarizes our experience educational program described in this paper. Section V contains the conclusions.

II. E-Learning Material in Electrical Engineering

The e-learning material developed within the INETELE project mentioned in the Introduction covers a large part of the subjects of electrical engineering. Here, by opening a very narrow window, some samples of the chapters designed and worked out by us will be treated to show the main features of the whole projects.

Each chapter has two parts. The part "For Class" includes relatively few information in each page, providing fast comprehension. It applies big capital letters and plenty of animations. It has audible support by speech and music. It is intended to be used mainly in class by the teacher.

The part "For Home" is similar to a textbook containing all the necessary explanations which are usually given by the teacher in class.

The control of the program is very simple. It contains a number of functions: The user can move from page to page, forward or backward, by hitting any of the keys and or from chapter to chapter by the keys and . The relevant explanations in "For Home" can be reached immediately by key NOTE. The user can initiate the animation and stop it at will by keys , , and and search by keywords in INDEX. At the bottom of the page, further keys help the users (Fig. 1). For better comprehension, interactive animations are included. They are mathematical and drawing "engines" where users can solve the equations and draw the results online.

In presenting some of the characteristic pages from the project, we are proceeding from the simple to the more sophisticated pages as far as the way of presentation and the services offered for the teacher and student are concerned. The contents of the pages shown below were chosen from a very simple material to push the limelight toward the way of presentation and the services. The contents are well known for the readers anyway. They do not have any special value here.

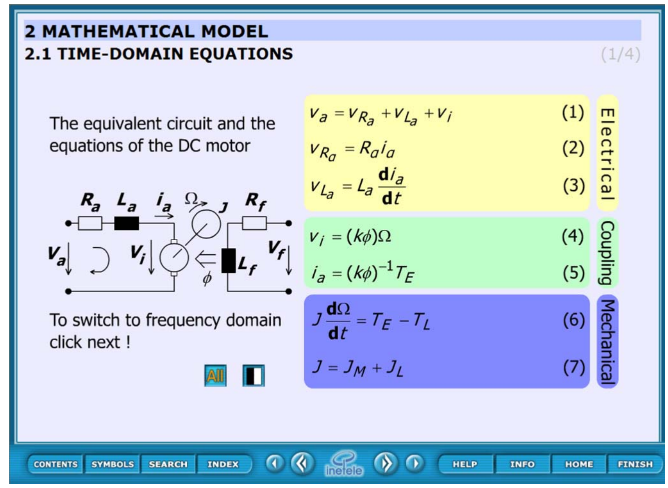

Fig. 1. Time-domain equations of a dc servomotor.

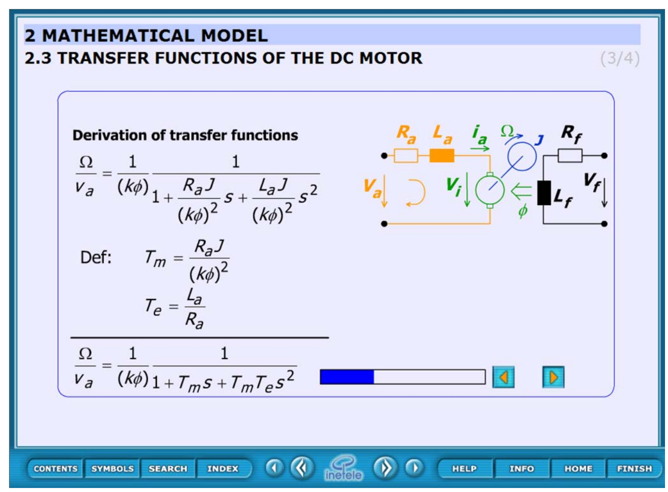

Fig. 2. Intermediate stage of manipulation with equations shown in Fig. 1.

The first examples treat the dc servomotor.

Fig. 1 shows the time-domain equations. In Fig. 2, the transfer function Ω/v_a is given, which is derived from the equations shown in Fig. 1. Here, Ω is the angular speed and v_a is the armature supply voltage. In the animated program, the two figures are examples for the *floating derivation technique*.

Starting from Fig. 1, a sequence of pages can be called step-by-step, showing the derivation of transfer function by manipulating, substituting, rearranging, etc., the equations in animated way as the teacher would perform it on the blackboard. The bar at the bottom of Fig. 2 shows how far the teacher proceeded in the manipulation of the equations from the final state.

Fig. 3 shows the static angular frequency Ω and torque T characteristics of the motor in the motoring domain.

In this interactive page, four variables can continuously be changed from zero up to a maximum value. The results are the changing static characteristic and the steady-state value of the angular speed.

Transformations of the block diagram of a dc motor are shown in Figs. 4–6 step-by-step. In this animation, the progress bar applied as well as in the previous case can be seen.

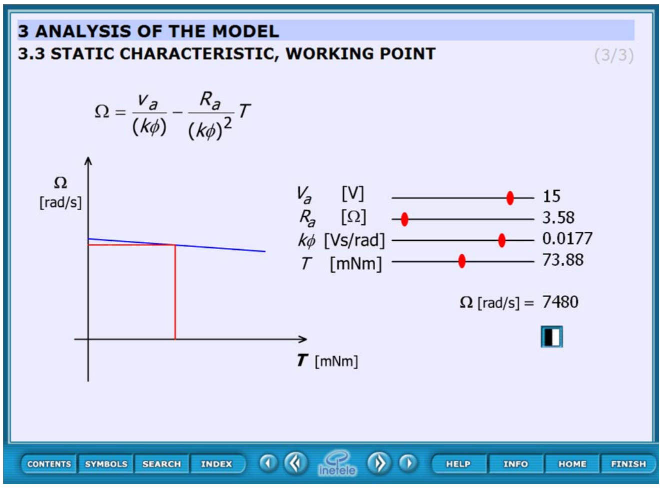

Fig. 3. Interactive page of the static torque–speed characteristics of dc servomotor.

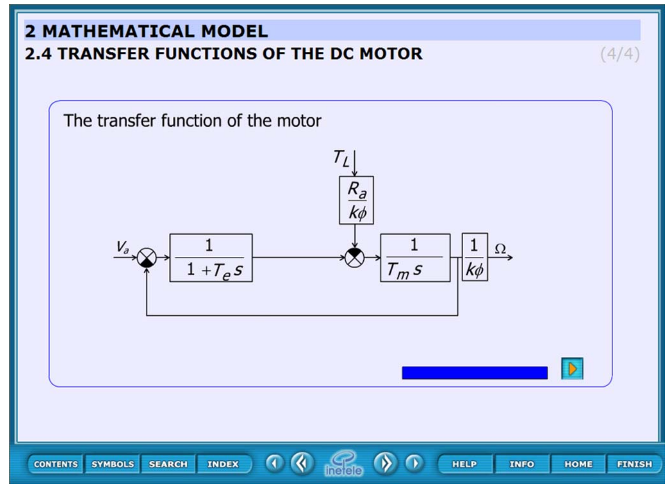

Fig. 6. Last screen of dc motor block diagram.

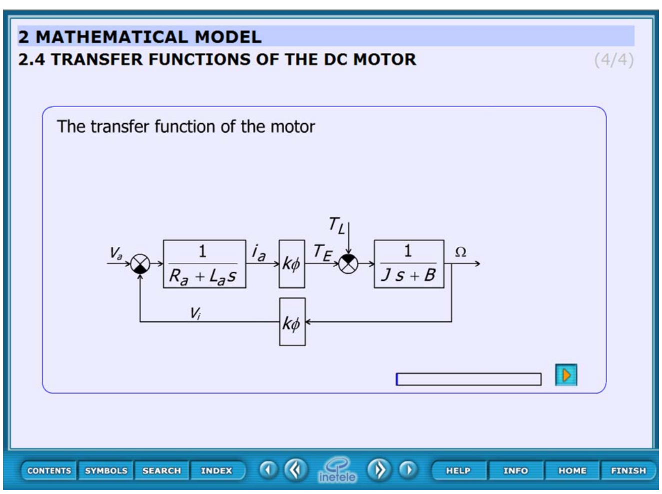

Fig. 4. First screen of dc motor block diagram.

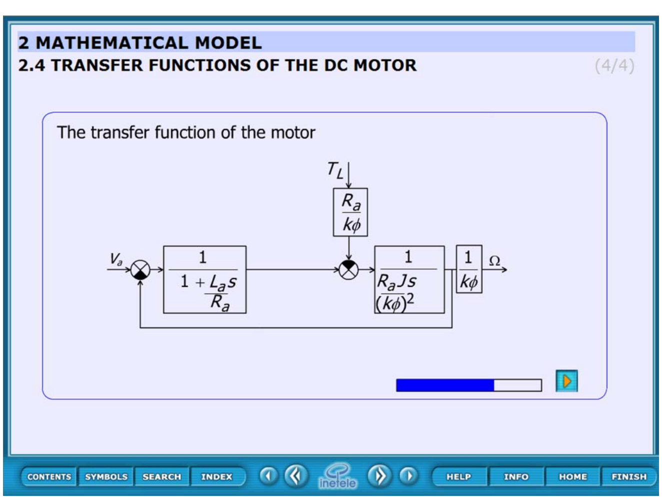

Fig. 5. Stage of the animation.

III. Internet-Based Laboratories

The user can learn from this measurement the control of a servo drive and robot by a computer. First, the user gets acquainted with the communication between the computer and the servo drive as the important component of a mobile robot. Next, the embedded system of a mobile robot is studied. (There is a separate manual for robot experiment.) We will use a D/A converter to send information from the computer to the servo drive since the reference signals are analog voltage signals in most servo drives. The movement of the motor is measured by an encoder, which sends impulse train to the computer. A counter counts the impulses of the encoder, and the computer can read the value of the encoder, informing it on the position of the motor. The speed of the motor is calculated from the actual and the previous position information. This method results in a very noisy speed signal. The measurement uses a discrete time filter to reduce this measurement noise. The user can learn the way of writing and tuning the PI controller for this simple servo drive. The user does not need to be expert in computer programming, but she/he has to know the basics. The user will write some very simple program in C language and Visual Basic. Examples are shown, and it is believed that the user can carry out this measurement even if it will be her/his first C or Visual Basic program.

In traditional laboratory exercises, small student groups (two to five people) are running experiments in a given time frame on one measurement station. They solve the problem, use the system, and write measurement reports together, hiding each individual's effort and level of understanding on the given topic. However, the experience earned during these exercises is the most valuable part of the engineering education; this is the way how students learn to solve problems and use different equipment. The more students running the same measurement at the same time, the less effective it will be. Remote laboratories operate by the opposite principle: Focus on the individual but disconnect the student from the physical equipment. The overall goal is to provide the same work conditions as the measurement would be run in the traditional way. The student's eye is usually replaced by a Web camera facing the measurement system, and the equipment is replaced by graphical user interfaces running on the student's own PC. Based on the different graphical user interface types, the remote laboratories can be categorized in the following way.

1) *Dedicated client application*. The remote laboratory experiments can be accessed by a dedicated client application, which runs on the student's PC (e.g., [8]–[10]).
2) *Lightweight Web browser*. The built-in Web browser without any special add-on (e.g., ActiveX and LabView) is used to access the experiment server (e.g., [11]).
3) *Web browser with dedicated add-ons*. A Web browser is used as a frame to run a dedicated add-on (e.g., LabView), which connects to the experiment server (e.g., [12]–[16]).
4) *Hybrid solutions*. Combination of the aforementioned categories (Web browser is used to sign up for experiments, but a dedicated application is needed for running the experiments) (e.g., [17]–[19]).

Remote laboratories can be further categorized based on the interaction types with the measurement station.

1) *Online measurements*. Experiments are executed in real time on the measurement system. There is a possibility for parameter modification or program upload (e.g., [20], [21]).
2) *Offline measurements*. Experiments are prerecorded; students are in a virtual laboratory (e.g., [22], [23]).

Within our project, only the online lightweight Web-browser-based laboratory test is applied, and only its discussion is included and treated below. The proposed remote laboratory allows students to write different types of controller for a servo motor, which is connected to the measurement server. The measurement server is hosting a homepage, where students can upload and execute their controller code (written in C). The biggest advantage is the full freedom in controller definition. There is no restriction in the amount and type of parameters. One of the most important contributions in the following approach is that, after only a very short learning period, the students are able to create various types of controllers. All the controllers (P, PI, PID, sliding mode, etc.) can be realized, tested, and observed by them, and every little change can be seen in the results. The students can even see (via a Web camera) what they are doing in the laboratory.

A. Requirements

The following requirements were set against the remote laboratory.

1) All experimental equipment should be available 24 h a day (anytime).
2) Students can reserve time slots for running experiments (access restriction).
3) The system should handle a maximum of 40 students at a time (robustness).
4) No or minimal previous programming skills are needed (will be explained later on).
5) No external add-on (besides Web browser) is needed for use (platform independent, fast access).

B. Applied Software Tools

Our approach is a standard Web-browser-based remote laboratory but with large flexibility in controller design and in selection of system supply and load. The Web site does not need any add-on, because all the logics and intelligence are located on the server side, which results in thin client side. The server side is implemented in ASP.NET in a multitasking manner. Each student accessing the measurement Web site is identified by his/her own ID, composed of the IP address, measurement number, and the Web browser's details. This ID is then used in each result generation and measurement execution, eliminating the problems rose by concurrent access. Depending on the measurement task, the students write their own program in C (headers and basic variables are provided beforehand) or write only the missing code into a program skeleton. This reduces the amount of needed program code for each measurement task to an average of 10–20 new rows. The most important difference, compared to existing remote laboratories in [24]–[27], is that, in our case, the students are not limited to modifying or giving parameters to control programs; they can create their own state variables, even their own control solution. The queuing is solved in the similar way; each student measurement execution is waiting in a row, working in the first-in first-out order. This is shown in Fig. 7.

In order to ensure the measurement equipment's long lifetime, the equipment is only turned on when there is a request from the student Web browser. This is solved by utilizing a simple power supply unit connected with one end to the D/A card output channel and the other end to the measurement equipment. As the system operates on standard Windows environment, which does not guarantee real time in any interrupt or timer event occasion, an extension is needed. A software solution is used (provided by IntervalZero): It is possible to create timers with 100-μs interval, with a maximum of 0.001-ns yield. Not only that the running environment is created but also it seamlessly integrates itself to C++ compilers, which allows running programs without modification in real-time environment. With this solution, more user-friendly programming of real-time applications can be achieved. The usability of the real-time extension has been proven in industrial applications.

The hardware architecture of the remote laboratory is shown in Fig. 8.

Each exercise can be reached through the homepage of the experiment address. The order of the exercises is defined in such a way that helps the students to build them on the results of the previous exercises. The layout of the homepage can be seen in Fig. 9.

In order to control the dc motor connected to PC, the following system components performing basic functions are needed:

1) a card generating analog output signals to turn on/off the drive and generate the reference signal (e.g., D/A card);
2) a card measuring input encoder signals, forwarding it to the PC (e.g., A/D card or counter card);
3) a real-time clock that can schedule signal sampling or task execution.

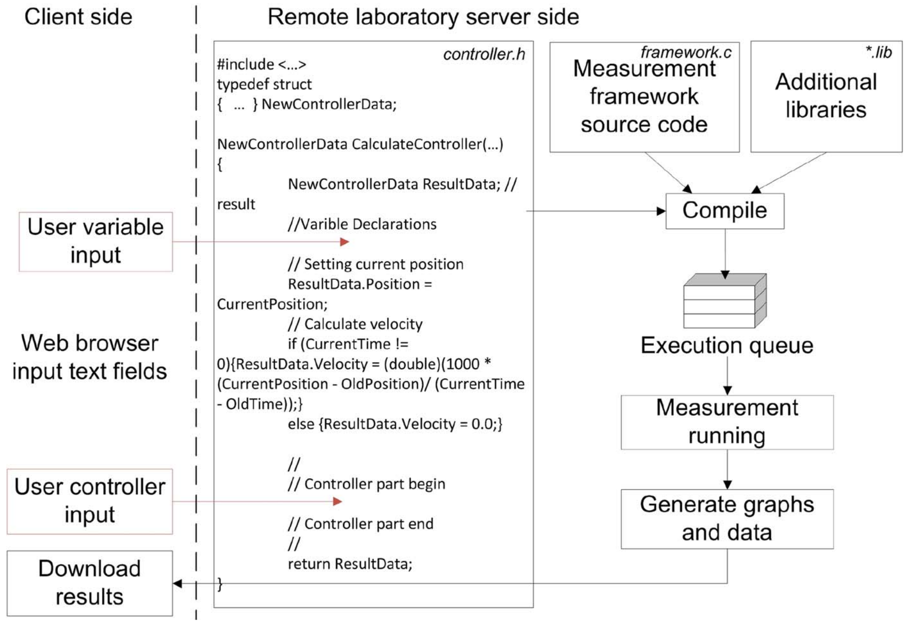

Fig. 7. Controller code upload, compile, and execution in the remote laboratory.

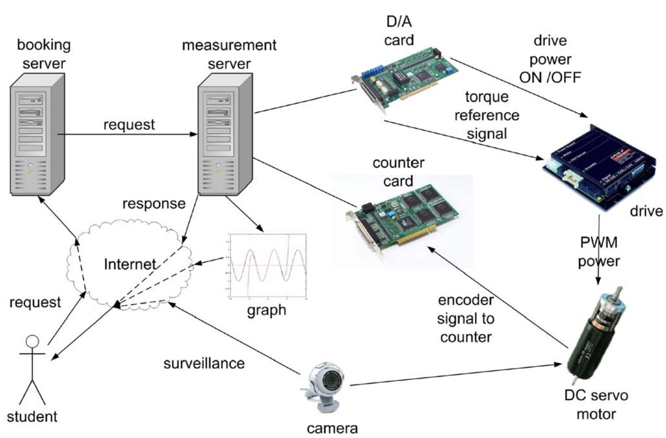

Fig. 8. Hardware architecture of remote laboratory.

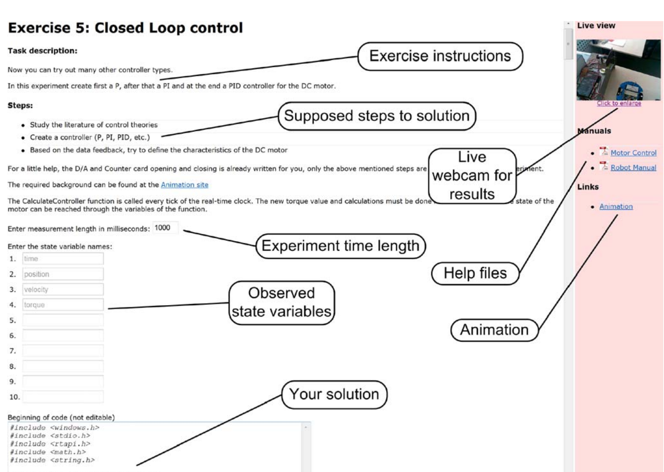

Fig. 9. Measurement homepage layout.

The students will start learning from the basics of programming of aforementioned cards through implementing modern control theories in real circumstances.

The aim of the remote laboratory is that the student can learn by these measurements the control of a servo drive by a computer. First, the student gets acquainted with the communication between the computer and the servo drive. The student does not need to be an expert in computer programming, but she/he has to know the basics. The student will write some very simple program in C language. Examples are shown, and it is believed that the student can carry out this measurement even if it will be her/his first C program.

C. Hardware Elements

The key system components applied here are the following:

1) industrial Siemens PC with Pentium-4 2.8-GHz processor;
2) Advantech PCI-1720 D/A output card;
3) Advantech PCI-1784 counter input card;
4) servo drive including
   a) servo amplifier;
   b) Maxon A-max 26(110961) dc motor;
   c) Maxon Digital Encoder HP HEDL 5540;
   d) Maxon Planetary Gearhead GP 26(110395);
5) webcamera for visualizing the laboratory;
6) MATLAB programs.

D. Time Slot Reservation

The students first have to reserve a time slot for the measurements before they can start the exercises. The authentication and access control is managed by this time reservation server (further on booking server). Every student who will participate in the remote laboratory exercises needs to have a username and a password in order to access the time reservation tables. If there are free "virtual seats" for the given measurement, the students can sign up for a given time slot. The amount of "virtual seats" depends on the settings for the given measurement type (how many parallel measurements can run at the same time) and the time of the day (e.g., weekdays or weekends can differ). In the given time slot, the measurement homepage can be reached from the booking server via a special hyperlink. To establish a common platform for remote laboratories, a centralized third-party booking server is used, which is based on Moodle [3].

E. Measurements

In the remote laboratory exercise, some of the relevant parts of the multimedia e-learning program package worked out in the INETELE project containing animations, interacting pages, Flash movies, etc., are incorporated. Some of the features of a remote laboratory for testing a servo system are summarized next. A step-by-step learning program explains to the students the background knowledge from the basics (computer-based measurements) to higher level (sliding mode control). In addition, the step-by-step learning program consists of the following lessons.

1) Exercise: Using the PCI-1720 D/A card (turning on/off the experimental system).
2) Exercise: Using the real-time clock with PCI 1720 D/A card (using the embedded real-time clock).
3) Exercise: Using the PCI-1784 Counter card (counting feedback from the experimental system).
4) Exercise: Open-loop measurement.
5) Exercise: Closed-loop measurement (designing and implementing a P, PI, PID, and sliding mode controller).

F. Exercises

The first three exercises seem to be very easy, and they were not included in the first solution. Later, we recognized that we cannot skip these simple exercises. Many of the students are usually fulfilling their M.Sc. studies in mechanical engineering or B.Sc. students in mechatronics and not well trained in computer programming. This is mainly because of their study plans. These students, during their first year, are introduced (total of 6 study points, which is equal to 180 h of study) to different types of programming languages (e.g., Visual Basic and C++), but as they are not using this knowledge in every day basis, such as computer science students, their experience is low and cannot solve independent programming tasks. For example, in exercise 1, the students have to find two appropriate program lines of the sample program of D/A controller where they have set (write) the channel number and the output voltage value only. We could not imagine how it is going to be a big step for a student of mechanical engineering when he/she can switch on an electrical unit by computer in the first time in his/her life.

All exercises include two to four tasks, and the students have to add a maximum of ten new lines to the given framework code in each task. The framework code includes the driver programs of D/A and counter cards (i.e., the results of the first three exercise). However, the students do not need to retype those codes; they can focus on the controller parts.

The first real measurement is exercise 4, which is an open-loop experiment. More advanced measurements can be run in

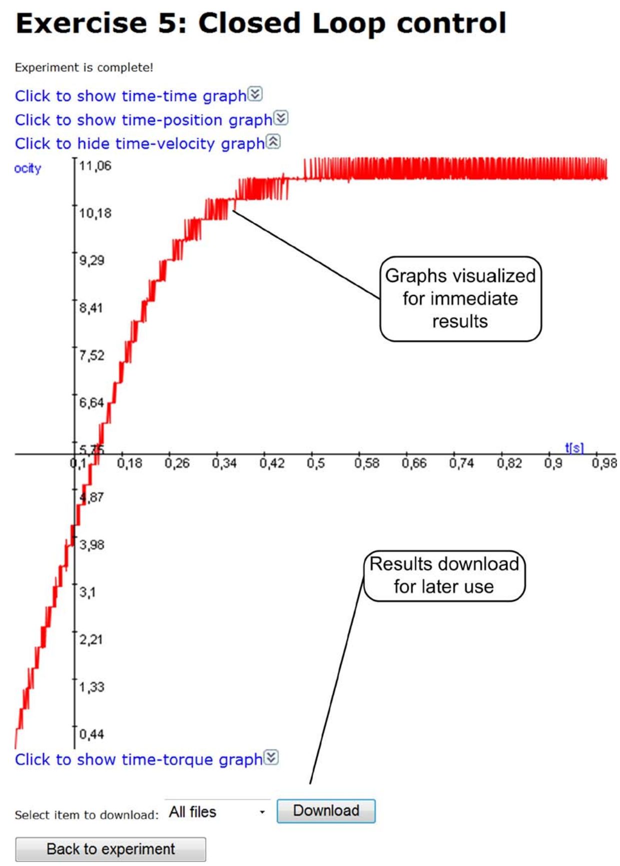

Fig. 10. Result download page, raw data of the velocity ([rad/sec]). This is generated automatically right after executing a measurement to plot the raw data. It is considered as an indicator, which shows whether the measurement is acceptable and whether the results are worth downloading.

exercise 5. In the following, two different tasks (one beginner and one advanced) will be introduced.

Task 1 (Belongs to Exercise 4): In this exercise, open-loop measurements will be performed with the dc motor. In the first task, the electrical torque is increased from $0\ \mathrm{mN \cdot m}$ *to the maximum torque of the motor* ($700\ \mathrm{mN \cdot m}$). *This way, the step responses of various motor variables to different step change sizes in torque signals can be obtained. The result files, which can be downloaded at the end of the measurement, can be evaluated in Matlab. There is a program already written for this task. Running this program results in the shaft speed–time, torque–time*, and *position–time diagrams. These diagrams can also be saved in jpg format for further documentation.*

The code, which is written by the student, is a single line

$$ResultData.Torque = 0.7 \text{ (The Torque is given in } \mathrm{N \cdot m}).$$

This is inserted to a framework system, compiled and executed. This means that the torque of the motor will be $0.7\ \mathrm{N \cdot m}$. The servo drive will reach a specific rotation speed when the electric torque will be equal to the load torque.

After error checking and compiling, the experiment is run, and the results can be downloaded.

Every measurement ends with a result page (see Fig. 10), where the experiment results can be seen and downloaded in

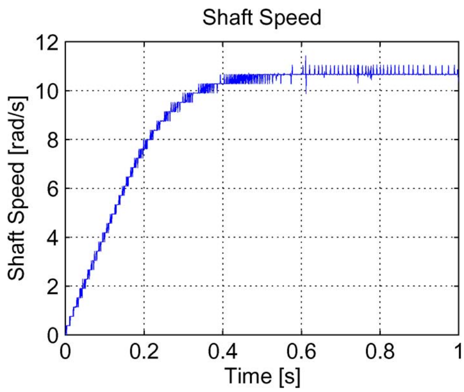

Fig. 11. Experiment result (time function of angular velocity).

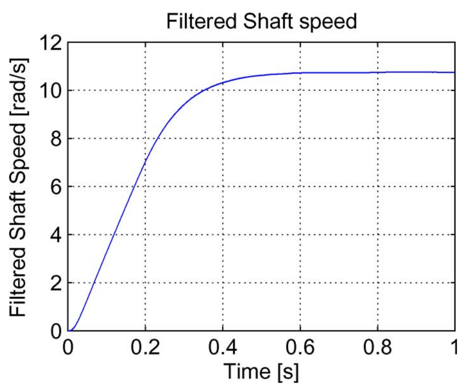

Fig. 12. Experiment result with filtered Ω (time function of angular velocity).

the following ways:

1) plots of the time functions of the selected state variables, for fast visualization;
2) data files (Matlab files) of the selected state variables, for download and later use;
3) implemented controller, for assigning the controller to the result;
4) all results in one compressed file, for handing in experiment results.

The result files, which can be downloaded at the end of the measurement, can be evaluated in Matlab. There is a MATLAB program, which can be downloaded for plotting the results. Running this MATLAB program results in the shaft speed–time, torque–time, and position–time diagrams. The students can use MATLAB at the university, and they can modify the plots. The resulting state variable files are simple text files, which are readable by humans or any software tool. The results can be downloaded in jpg format as well (if a student cannot use MATLAB at home). Of course, the graphic files cannot be modified.

The result of this task can be seen in Fig. 11 (plot of angular velocity). It is very noisy. A smooth curve can be obtained from the noisy signal of Ω by low-pass filtering (see Fig. 12) in the following way:

$$\Omega_{\text{filtered}} = T_c^3 \Omega_{\text{filtered}}^{(3)} + 3T_c^2 \Omega_{\text{filtered}}^{(2)} + 3T_c \Omega_{\text{filtered}}^{(1)} + \Omega_{\text{measured}} \tag{1}$$

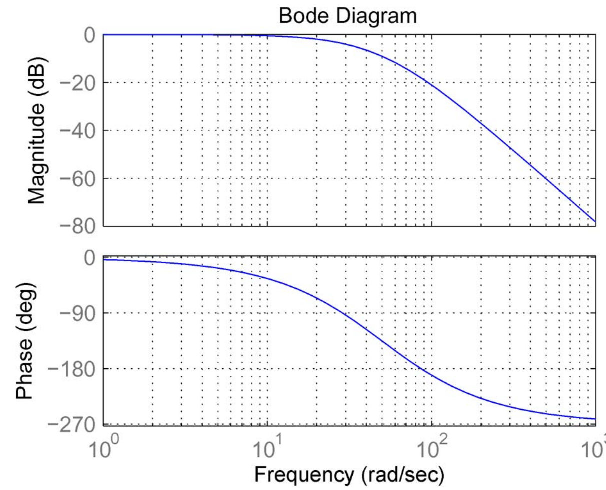

Fig. 13. Bode plots of the filter (2) in the case of $T_C = 0.02$.

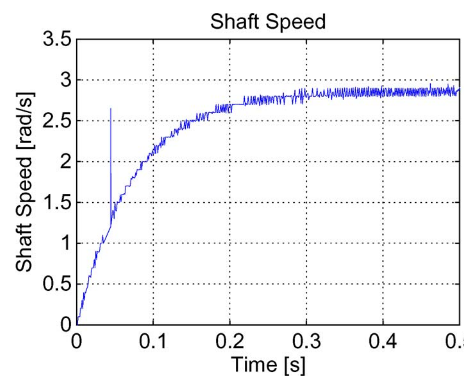

Fig. 14. Response of P-type controller with $P = 10$.

where T_C is the time period of the cutoff frequency and $^{(i)}$ is short for ith derivate with respect to time. The state space equation of the filter (1) is

$$\dot{x} = Ax + By \tag{2}$$

where

$$x = \begin{pmatrix} \Omega_{\text{filtered}} \\ \Omega_{\text{filtered}}^{(1)} \\ \Omega_{\text{filtered}}^{(2)} \end{pmatrix} \quad A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\frac{1}{T_c^3} & -\frac{3}{T_c^2} & -\frac{3}{T_c} \end{pmatrix}$$

$$B = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{T_c^3} \end{pmatrix} \qquad y = \Omega_{\text{measured}}. \tag{3}$$

The state-space equation (2) can be rewritten in discrete form and calculated for a given sampling time T_{sample} and cutoff frequency by c2d (continuous to discrete) command of MATLAB (the transfer function is shown in Fig. 13)

$$[Ad, Bd] = \text{c2d}(A, B, T_{\text{sample}}). \tag{4}$$

In Figs. 14 and 15, two responses of a P-type controller are compared. The students have to declare three new variables

$$float\ P = 10 \text{ or } P = 30$$

$$float\ ref\_vel = 3$$

$$float\ error\_vel = 0$$

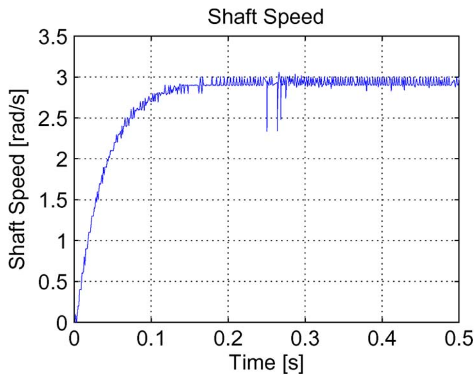

Fig. 15. Response of P-type controller with $P = 30$.

and they have to write two new lines into the controller implementation part

$$\begin{array}{r}error\_vel = ref\_vel - ResultData.Velocity\\ ResultData.Torque = P * error\_vel\end{array}.$$

Task 2 (Belongs to Exercise 5): In this exercise, closed-loop measurements will be performed with the dc motor. The task is to design a sliding mode controller for the position of the motor [28]. *The constant reference position is set to*

$$\alpha_{\text{ref}} = 10 \text{ rad}.$$

The students can learn the design steps of a sliding mode controller from the free animation. The position and speed error are defined as

$$\begin{aligned}\alpha_{\text{error}} &= \alpha_{\text{ref}} - \alpha_{\text{motor}}\\ \Omega_{\text{error}} &= 0 - \Omega_{\text{motor}}.\end{aligned}$$

The sliding surface is selected as

$$\sigma = \lambda\alpha_{\text{error}} + \Omega_{\text{error}}.$$

The control law is selected as

$$torque = 0.1 \text{ sign}\sigma.$$

The exercise results are available in the same format as in the case of Task 1.

The code, which is written by the student, is divided into two sections: variable declaration and controller implementation. Only the filtered solution is presented here:

Variable declaration

$$\begin{array}{l}float\ sigma,\ error,\ error\_dot\\ float\ ref = 10.0\\ float\ lambda = 0.5\\ static\ float\ xd\_1 = 0.0,\ xd\_2 = 0.0,\ xd\_3 = 0.0\\ static\ float\ xtmp\_1 = 0.0,\ xtmp\_2 = 0.0\\ /^*\ discrete\ time\ filter\ variables\ (4)\ in\ case\ of\\ \qquad T_c = 0.02\ and\ T_{\text{sample}} = 0.001\ ^*/\\ float\ Ad11 = 1,\ Ad12 = 0.0010,\ Ad13 = 0.0000\\ float\ Ad21 = -0.0005,\ Ad22 = 0.9999,\ Ad23 = 0.0010\\ float\ Ad31 = -0.9851,\ Ad32 = -0.2960,\ Ad33 = 0.9703\\ float\ Bd1 = 0.0000,\ Bd2 = 0.0005,\ Bd3 = 0.9851.\end{array}$$

Controller implementation

$$\begin{array}{l}/^* filter\ ^*/\\ xtmp\_1 = Ad11 * xd\_1 + Ad12 * xd\_2 + Ad13 * xd\_3\\ \qquad + Bd1 * ResultData.Velocity\\ xtmp\_2 = Ad21 * xd\_1 + Ad22 * xd\_2 + Ad23 * z\_3\\ \qquad + Bd2 * ResultData.Velocity\\ xd\_3 = Ad31 * xd\_1 + Ad32 * xd\_2 + Ad33 * xd\_3\\ \qquad + Bd3 * ResultData.Velocity\\ xd\_1 = xtmp\_1\\ xd\_2 = xtmp\_2\\ error = ref - ResultData.Position\\ error\_dot = -xd\_1\\ sigma = lambda * error + error\_dot\\ ResultData.StateVariable\_5 = sigma\\ ResultData.StateVariable\_6 = xd\_1\\ if(sigma > 0)\\ \{ResultData.Torque = 0.1\}\\ else\ if\ (sigma < 0)\\ \{ResultData.Torque = -0.1\}\\ else\ if\ (sigma = 0)\\ \{ResultData.Torque = 0\}.\end{array}$$

The results can be seen in Figs. 16–19.

IV. Educational Experience

In a normal laboratory experiment, one to three students can use one experiment setup. In our case, one teacher can sit in a computer room with approximately 30 students to find out what should be written in a control program and where it should be written. It would take up to 1 h, and the overall time used by the experimental system is only a few minutes. (In this type of experiment, one test is running for 1–10 s.) After learning the usage of this software, the students can examine the system at home as long as they want. Around 40 students can log in at the same time. This already shows the efficiency of the remote laboratories.

The remote laboratory system was first used during the 2008/2009 autumn semester, and over 20 students were using it (third-year B.Sc. mechanical engineering students). The remote laboratory served as an additional course material to traditional book and lecture-based tutoring. At the end of the course, students filled out a questionnaire, where they could grade the usability, structure of homepage, accessibility, etc. In the following, some of these answers will be discussed. In general, the students liked the idea of having a remote laboratory available 24 h a day; however, they were arguing for more measurements besides the dc motor. The animation and simulation materials were reported to be useful, particularly among those students who could not attend every class. The first three exercises listed in Section III-E were not preferred by the students, as they thought that the exercises involve a lot of typing. On the other hand, it was a real breakthrough for most of them as it was the first time when they could turn on–off an external unit which is not a part of the computer. The user manuals for the exercises were reported to be well structured

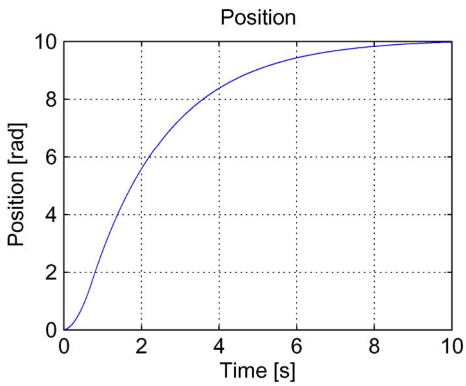

Fig. 16. Experiment result (position).

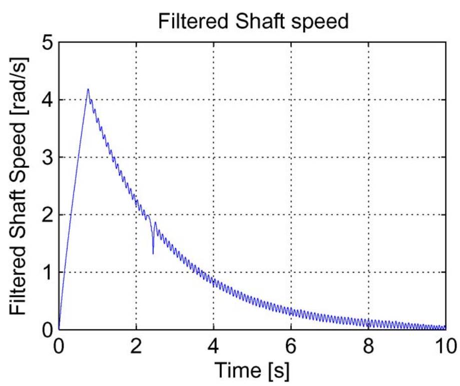

Fig. 17. Experiment result (Ω filtered angular velocity).

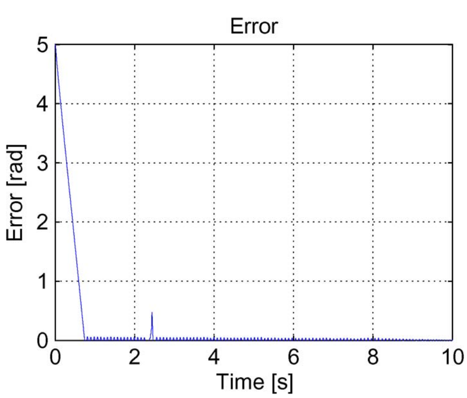

Fig. 18. Experimental result (error).

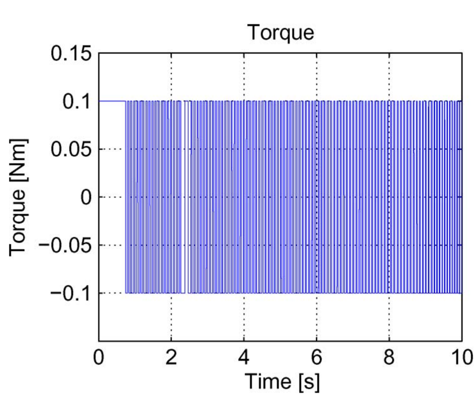

Fig. 19. Experimental result (torque).

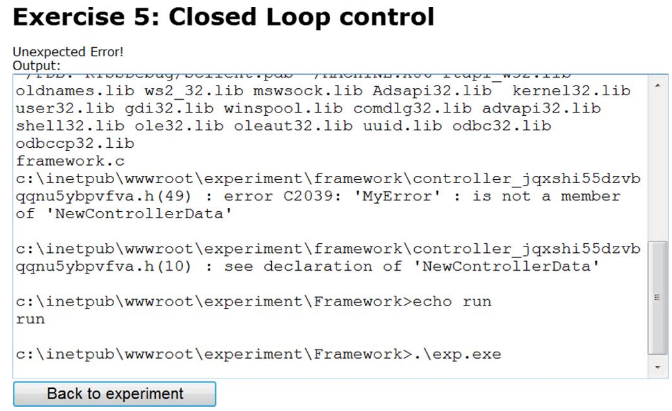

Fig. 20. Mistyped variable name in the controller code returns this error prompt.

and clear in content, providing enough help for accomplishing the exercises. Most of the comments given were related to the program understanding/writing parts of the exercises, in details.

1) Lack of programming skills/background: During the exercises, C language syntax is used; this could be replaced by a more user-friendly solution (the input text is first transformed to C language and executed afterward, instead of direct C language input).
2) Error messages are the same as in C language and are not recognizable for the user: For example, a mistyped variable name in the controller part $(ResultData.MyError = 12.0;)$ results in the following error, as an output in a textbox (see Fig. 20).:
3) Text-based input should be replaced by other type of inputs: Assigning variable values could be done by side bars or knots.

Based on the feedback, the remote laboratory is forward developed and is continuously used in B.Sc. and M.Sc. education. It can be visited at: http://dind.mogi.bme.hu/experiment/.

V. Conclusion

A complete distance learning program has been presented starting from the animation and ending at Internet-based measurement. It is difficult to describe a dynamic and interactive animation by static screens. The problem is the same when the operation of a motor must be explained by static figures. Because of the separation of the communication frame program and the controller program, all students can carry out this simple measurement even if they are not good at programming. Since the actual measurement when the motor is allocated to one student takes only a few seconds, several students can measure the same motor virtually at the same time. The development of such educational program needs a complete new way of thinking; some elements of this new educational approach have been presented in this paper.

References

[1] T. Wolbank, P. Bauer, P. Macheiner, and M. Vogelsberger, “Distance laboratory for teaching industrial electronics,” in *Proc. 34th IEEE IECON*, Nov. 2008, pp. 3497–3502.
[2] P. Bauer, V. Fedak, and K. G. van den Biggelaar, “Application of elearning in electrical engineering,” in *Proc. 6th Int. Conf. Emerging eLearning Technol. Appl.*, Sep. 2008, pp. 7–15.

[3] D. Hercog, B. Gergic, S. Uran, and K. Jezernik, "A DSP-based remote control laboratory," *IEEE Trans. Ind. Electron.*, vol. 54, no. 6, pp. 3057–3068, Dec. 2007.
[4] P. Bauer, B. Davat, V. Hajek, and V. Fedak, "Educational visualization for teaching power electronics," in *Proc. IEEE Workshop Power Electron. Edu.*, Jun. 2005, pp. 45–51.
[5] J. Dudrik, "Education of power electronics devices using e-learning," in *Proc. 3rd ICETA*, Sep. 2004, pp. 427–430.
[6] V. Fedak and P. Bauer, "E-learning in education of electrical drives and power electronics: Opportunities and challenges," in *Proc. Int. Conf. EDPE*, Sep. 2005, pp. 1–9.
[7] J. Dudrik, "Examples from power electronics-examples using pspice programme," in *Proc. Int. Workshop Courses Microcomput.-Control. Elect. Drive Syst. Ind. Autom.*, 1994, p. 51.
[8] R. Marques, J. Rocha, S. Rafael, and J. F. Martins, "Design and implementation of a reconfigurable remote laboratory, using oscilloscope/PLC network for www access," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2425–2432, Jun. 2008.
[9] C. Lazar and S. Carari, "A remote-control engineering laboratory," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2368–2375, Jun. 2008.
[10] A. Bagnasco, A. Boccardo, P. Buschiazzo, A. Poggi, and A. M. Scapolla, "A service-oriented educational laboratory for electronics," *IEEE Trans. Ind. Electron.*, vol. 56, no. 12, pp. 4768–4775, Dec. 2009.
[11] M. Wu, J.-H. She, G.-X. Zeng, and Y. Ohyama, "Internet-based teaching and experiment system for control engineering course," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2386–2396, Jun. 2008.
[12] W. Hu, G.-P. Liu, D. Rees, and Y. Qiao, "Design and implementation of web-based control laboratory for test rigs in geographically diverse locations," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2343–2354, Jun. 2008.
[13] V. F. Pires, L. S. Martins, T. G. Amaral, R. Marcal, R. Rodrigues, and M. M. Crisostomo, "Distance-learning power-system protection based on testing protective relays," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2433–2438, Jun. 2008.
[14] A. Leva and F. Donida, "Multifunctional remote laboratory for education in automatic control: The CrAutoLab experience," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2376–2385, Jun. 2008.
[15] L. Costas-Perez, D. Lago, J. Farina, and J. J. Rodriguez-Andina, "Optimization of an industrial sensor and data acquisition laboratory through time sharing and remote access," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2397–2404, Jun. 2008.
[16] M. T. Restivo, J. Mendes, A. M. Lopes, C. M. Silva, and F. Chouzal, "A remote laboratory in engineering measurement," *IEEE Trans. Ind. Electron.*, vol. 56, no. 12, pp. 4836–4843, Dec. 2009.
[17] M. J. Callaghan, J. Harkin, T. M. McGinnity, and L. P. Maguire, "Intelligent user support in autonomous remote experimentation environments," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2355–2367, Jun. 2008.
[18] S.-C. Wang and Y.-H. Liu, "Software-reconfigurable e-learning platform for power electronics courses," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2416–2424, Jun. 2008.
[19] L. L. Bello, O. Mirabella, A. Raucea, and L. Capetta, "ENEL PILOT: From a research testbed to a virtual educational laboratory," *IEEE Trans. Ind. Electron.*, vol. 56, no. 12, pp. 4844–4853, Dec. 2009.
[20] H. Hassan, C. Dominguez, J.-M. Martinez, A. Perles, and J. Albaladejo, "Remote laboratory architecture for the validation of industrial control applications," *IEEE Trans. Ind. Electron.*, vol. 54, no. 6, pp. 3094–3102, Dec. 2007.
[21] L. Indrusiak, M. Glesner, and R. Reis, "On the evolution of remote laboratories for prototyping digital electronic systems," *IEEE Trans. Ind. Electron.*, vol. 54, no. 6, pp. 3069–3077, Dec. 2007.
[22] M. Casini, D. Prattichizzo, and A. Vicino, "The automatic control telelab—A web-based technology for distance learning," *IEEE Control Syst. Mag.*, vol. 24, no. 3, pp. 36–44, Jun. 2004.
[23] A. R. S. Castellanos, L. Hernandez, I. Santana, and E. Rubio, "Platform for distance development of complex automatic control strategies using MATLAB," *Int. J. Eng. Edu.*, vol. 21, no. 5, pp. 790–797, 2005.
[24] C. Gravier, J. Fayolle, B. Bayard, M. Ates, and J. Lardon, "State of the art about remote laboratories paradigms—Foundations of ongoing mutations," *Int. J. Online Eng.*, vol. 4, no. 1, pp. 19–25, Feb. 2008.
[25] P. Bauer, J. Dudak, D. Maga, and V. Hajek, "Distance practical education for power electronics," *Int. J. Engng Ed.*, vol. 23, no. 6, pp. 1210–1218, 2007.
[26] L. Gomes and S. Bogosyan, "Current trends in remote laboratories," *IEEE Trans. Ind. Electron.*, vol. 56, no. 12, pp. 4744–4756, Dec. 2009.
[27] J. Garcia-Zubia, P. Orduna, D. L. de Ipina, and G. R. Alves, "Addressing software impact in the design of remote laboratories," *IEEE Trans. Ind. Electron.*, vol. 56, no. 12, pp. 4757–4767, Dec. 2009.
[28] P. Korondi and H. Hashimoto, "Sliding mode design for motion control," in *Studies in Applied Electromagnetics and Mechanics*. Amsterdam, The Netherlands: IOS Press, 2000, pp. 221–232.

Gábor Sziebig (S'07–M'08) received the M.S. degree in technical informatics from Budapest University of Technology and Economics, Budapest, Hungary, in 2007. He is currently working toward the Ph.D. degree at Narvik University College, Narvik, Norway.

His research interests include robotics, manufacturing, teleoperation, vision systems, and cognitive communication channels.

Béla Takarics (S'08–M'09) received the M.S. degree in mechanical engineering from Budapest University of Technology and Economics, Budapest, Hungary, in 2007, where he is currently working toward the Ph.D. degree.

His research interests include tensor product transformation-based control, sliding-mode control, e-learning, and robotics.

Péter Korondi (M'98) received the Dipl. Eng. and Ph.D. degrees in electrical engineering from the Technical University of Budapest, Budapest, Hungary, in 1984 and 1995, respectively.

Since 1986, he has been with Budapest University of Technology and Economics (formerly the Technical University of Budapest). From April 1993 to April 1995, he worked in the laboratory of Prof. Harashima and Prof. Hashimoto at the Institute of Industrial Science, The University of Tokyo, Tokyo, Japan, where he continues to spend a month each year, working on joint research. As a result of this cooperation, the Intelligent Integrated System Japanese–Hungarian Joint Laboratory was founded in 2001. He is also with the Computer and Automation Research Institute, Budapest. His research interests include telemanipulation and motion control.

Dr. Korondi is a founding member of the International PEMC Council, which became a chapter of the European Power Electronics Association.