Experiments with Inverted Pendulum: from Simulations to Remote Control

Katarína Žáková

Abstract**—** The paper is focused on a typical nonlinear system of the inverted pendulum plant that is used as a benchmark in control education at many universities. Except of overview of various approaches to the experimental work with the considered device, our attention was dedicated to one way of its remote control. The introduced approach presents a client-server application that connects .NET technology and Macromedia Flash format.

Keywords**—** Inverted Pendulum, Remote Control, Server Application, Client Application, C++, Flash.

I. Introduction

THE necessity of learner experimental work in laboratories represents one important aspect of engineering education. Students solve practical problems and gain experience and practice needed for their future career. Introduction of real systems increases transparency of the solved examples. The main motivation for using of plants in the educational process is clear physical “visibility” of the controlled dynamics, and also the necessity to exercise all design steps starting with the plant identification and ending with the evaluation of the control results achieved with the particular model.

In last years expansion of Internet enabled to influence the education process in area of automation, too. Applications available via Internet help students to design and verify their control algorithms and to solve particular problems and tasks. Since they are placed on Internet they are available to students also in their free time (not only during lessons and practical exercises) and their use is not limited only to laboratories.

The paper demonstrates one way how to use remote control for the inverted pendulum model. The inverted pendulum is one of the classical problems of control engineering. It is a well-known example of nonlinear, unstable control problem. The inverted pendulum is related to rocket or missile guidance, where thrust is actuated at the bottom of a tall vehicle.

II. System Description

The inverted pendulum consists of a thin rod attached at its bottom to a moving cart. Whereas a normal pendulum is stable when hanging downwards, a vertical inverted pendulum is inherently unstable, and must be actively balanced in order to remain upright, typically by moving the cart horizontally as part of a feedback system.

The plant (Fig.1) is represented by a revolving pendulum mounted on top of a moving base [15]. By a DC-motor, a toothed wheel, a toothed belt and a clutch, the moving base can be driven along a guiding bar over a length of approximately 1.5m such that the pendulum is stabilized in upright position at a preassigned position. The stabilization of the pendulum is accomplished by a digital controller. Based on measurements, the controller generates a suitable signal, which controls the DC-motor by an electronic drive. The measurements are the pendulum angle and the cart position obtained by incremental encoders.

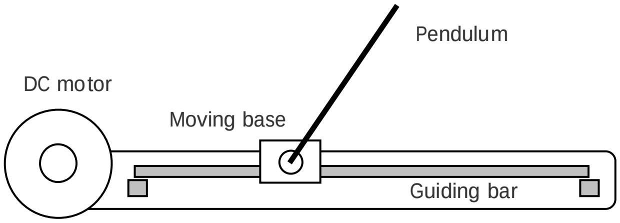


Fig. 1 Sketch model of inverted pendulum

At the controller design for this system, students usually start with control of the carriage whereby the pendulum hangs downwards. Then, they try to bring the pendulum in the upper position. After first experiments they find out that the task is not so easy. It is a good motivation for the choice of automatic control.

By its nonlinear dynamics, the inverted pendulum model gives an unique possibility to demonstrate basic features and limits of different linear and nonlinear control concepts. For this purpose at least two different approaches are usually applied. Mostly the control consists of 2 steps. The pendulum is swung up near to the desired vertical position where it has to be stabilized. Both activities require different control algorithms and therefore they are usually accomplished by 2

Manuscript received: March 31, 2007. Revised received: September 25, 2008. This work was supported in part by Slovak Grant Agency under Grant KEGA No. 3/3121/05 and Grant No. VEGA 1/3089/06. This support is very gratefully acknowledged.

K. Žáková is with the Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovičova 3, 812 19 Bratislava, Slovakia (e-mail: katarina.zakova@stuba.sk).

various controllers (energy controller, state space controller, fuzzy or neural controller, etc.).

For each particular design, the problems of the state reconstruction, measurement quantisation and control signal limitation are taken into considerations. Except of this the identification of an unstable nonlinear system (analytical identification combined with experimental identification of particular nonlinear terms by planning experiments – curve fitting, step response analysis requiring closed loop stabilisation of the starting position, measurement & quantization noise analysis) has to be done.

III. Possibilities of Experiment Realization

During the study students have several possibilities that enable them to become more familiar with the introduced plant. During the educational process they can use at least one of the following ways:

- Matlab or Maple simulation environment that except of control enables animation or 3D visualization of the model;
- local control of the real system;
- remote control of the real system.

A. Local Control – Simulations and System Animations

When engineering students start to solve the first problem based tasks they usually meet with computer simulations firstly. Only after they also continue with experiments on real plants. A computer simulation enables to simulate an abstract model of a particular system. Computer simulations have become a useful part of mathematical modeling of many natural systems in physics, chemistry and biology, human systems in economics, psychology, and social science and in the process of engineering technology, to gain insight into the operation of those systems. Many students can work e.g. with Matlab or Maple that are widely used at universities.

Matlab allows easy matrix manipulation, plotting of functions and data, signal and image processing, communications, control design, test and measurement, implementation of algorithms, creation of user interfaces, interfacing with programs in other languages, etc. Except of that, Matlab has available many add-on toolboxes that extend its environment to solve particular classes of problems in various application areas. The created simulation block scheme can be easily modified; students can use own parameters or change the whole structure of the controller. After running the simulation they can immediately see the result of their effort by means of a graphical dependence or even an animated model.

While Matlab was originally developed as a language specialized in numerical computing, Maple presents a computer algebra system that is oriented to symbolic solutions of given problems. Maple combines a programming language with an interface that allows users to enter mathematics in traditional mathematical notation. As Matlab, it also belongs to interpreted and dynamically typed programming languages.

Both programming environments (Matlab and Maple as well) also enable to design and create interactive graphical user interface models of plants that respond to actions taken by a student. Computer simulations have the advantage of allowing a student to make judgments, and also to make errors. The process of interactive learning through testing, evaluation, decision-making, and error correction creates the much stronger learning environment than passive instructions.

B. Local Control – Real Experiments

After short period of simulations students like to start to experiment with real plants. It brings many advantages. Students can work directly with the plant, they can touch it, they can hear the sound, they can immediately see the real response to their inputs. The control velocity depends mainly on the dynamics of the system and on the design of controller.

By real experiments student realize that control algorithm designed for the simulated simplified mathematical model may not work properly. It is necessary to take into account present noise, possible delays and unmodelled dynamics.

The presented model belongs to the systems directly controllable by PC in real time. The already mentioned Matlab environment also supports communication with real systems and therefore it can be used as a platform for control of educational plant models. In this way the system can be directly used for practical investigation of basic and advanced control engineering principles such as system dynamics modeling, identification, analysis and various control designs. The model is connected to the computer via an universal data acquisition card.

C. Remote Control – Real Experiments

In spite of the fact that experimental work in laboratory brings many advantages for students it is necessary to realize that the number of students is high in comparison with the number of available real plants. A possible solution of this problem is to support the education and it also means the access to laboratories via Internet.

Actually, the development of control applications on Internet is oriented in two directions. One group of developers tries to prepare virtual devices that by means of animations try to demonstrate the real situation and problems. In this case students can make own decisions also with errors without risk of any damage on the real device.

The second direction is devoted to the development of remote laboratories that enable student to attach to the real plant and to verify own control algorithms. The only disadvantage consists in the fact that student is not directly in the laboratory and he or she cannot touch the device by hands, to see it by own eyes, to hear the noise or to smell it. However, he or she receives all results that are needed for evaluation of experiment and therefore the presence in the laboratory is not really demanded. In addition, the big contribution is that in the moment of experiment student can be on the other side of the country or even the world and he or she can accomplish the experiment in any time.

IV. Requirements and Realization

For accomplishing the remote control of experiment it is necessary to create two applications: application for the server and application for the client.

The choice of programming tools for the preparation of the server side of experiment is usually influenced by several requirements:

- Expansibility of the experiment – the experiment should be easily spread without need for long study of the source code.
- Maximal use of system tools.
- Accessible hardware and software – it is necessary to take into account the most used systems and software environments at the universities and their accessibility for users.

The realization of the client side also requires considering various aspects of its design. It is needed to adjust requests of users, programmers and possible technical limits of application. The chosen tool should

- be suitable for animation.
- enable simple Internet communication.
- create a multi-platform application – the platform independence is important because the application is presented to users via Internet.

In the following section one possible solution for remote control of experiments is presented. We tried to choose software tools that respect all our requirements.

V. Direct Control of Remote Plant

As it was already mentioned Matlab package is one of the software environments that are very frequently used for solving of control engineering problems at universities over the whole world. Therefore our first idea was to use its capabilities not only for simulations and local control of experiments but also for their remote control. There exist several possibilities how to realize such experiment. Some of them are mentioned e.g. in [11]. However, in this approach we tried not to use Matlab engine since it is not available everywhere.

As it was already noticed, it was necessary to prepare both the server and the client applications.

For the server application we decided to use .NET technology since it is the product of the same company as the operating system that is installed on the computer taking care about the control of the inverted pendulum. This solution should provide the maximal compatibility between the operating system and the created application.

The client side we prepared using Macromedia Flash animation. This software enables to create nice animations that are independent on the platform. In addition, using ActionScript offers a simple control of application and Internet communication on the base of XML sockets.

Now, let us describe both applications in more details.

A. Server Side

The hierarchical structure of the server side is sketched in Fig.2. One can see there all-important classes that ensure the functionality of the server application.

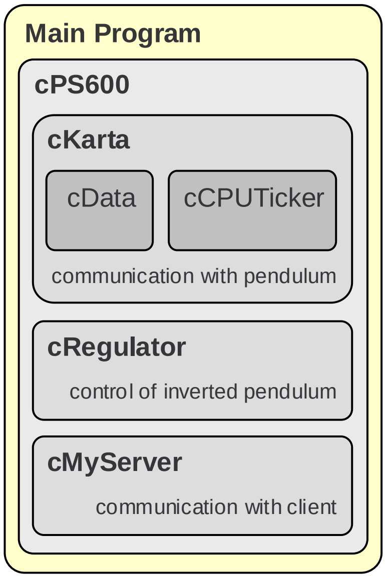

Main Program
cPS600
cKarta
cData
cCPUTicker
communication with pendulum
cRegulator
control of inverted pendulum
cMyServer
communication with client

Fig. 2. Hierarchical structure of the server classes

The class cData contains the structure for all important data such as position, angle, derivative of position and angle, set point and output that have to be transfer between threads and classes inside the server application. In spite of the fact that from programming point of view in this case a simple structure could be used, the implementation of the class brings one important advantage. Each time after a new instance of this object is created, its public internal variables are initialized to zero values. It simplifies work for programmer and gives higher security and stability to the code.

The class cKarta enables communication with the A/D card Option 600-3. It contains the header file and the library of functions (hudaqlib.lib) that are provided by the card producer Amira to the driver for Windows operating system.

The instance of the class cKarta occurs in the class cPS600 that presents the real model of the inverse pendulum. The class cPS600 guarantees exchange of data between the inverted pendulum (using access methods of the cKarta class) and the data using the class structure cData. The class cPS600 is responsible for example for the measurement of actual values from the device. In the given sampling period it is necessary to determine position of the pendulum, its angle and approximations of the position and angle derivatives. The measured and calculated values are saved to the instance of the cData class where the information about the required pendulum position and control signal is also placed.

The class cCPUTicker ensures the exact measurement of time. Despite the fact that under the Windows operating system the timing function exists in all C++ libraries, it often doesn't functions as it is expected. For example, the argument of the function Sleep() is an integer that represents the time period in milliseconds. However, at the real control of the system 10-millisecond time interval took sometimes 25 milliseconds of time. In this way it was not possible to guarantee the exact measurement of time and it was necessary to find a suitable alternative. The class cCPUTicker computes the processor ticks and using its methods one can determine the number of ticks that lapsed from the creation of the new instance of an object. The number of ticks can be easily converted to number of milliseconds.

The class cRegulator presents the controller. In this case we implemented the pole assignment controller. However, from the programming point of view the control algorithm can be easily changed or the user can be offered by several control algorithms in order he or she could have possibility to choose the right one. The only problem consists in the fact that the plant is control remotely and therefore it is necessary to ensure that all prepared control algorithms will be sufficiently safe. The inverted pendulum can rotate and move from one side to the other and it can happen that some person will be in the close vicinity of the plant when somebody else will approach to the system remotely. When the control algorithm is not safe and robust enough it can be danger. The bad designed control algorithm can also damage the plant itself.

The last class cMyServer covers the communication with the client. By means of this method the user receives the information about the pendulum position, its angle or about the actual settings. All requests are handled from the main program that has access to the plant data and also to the information that is or that will be sent to the client.

The main program is actually the instance of the native .NET class called Form. By the creation of this object the programmer initializes the whole equipment and sets all default values. The created user interface enables to control the plant and to visualize a feedback to the user. As it was already told the communication between the server and the remote client is realized using the cMyServer class.

The created program is realized by several threads (Fig.3) that enable parallel processing of several tasks:

- *Activation of the power converter.* The output of the A/D card Option 600-3 gives only the low-voltage unified signal ±10V. Since the used DC motor can work with much better performance that the card can supply, the provided signal has to be amplified.
- *Detection of the remote experiment state.* If the control is running, the value of the control signal is computed and it is sent to analogue output of the A/D card. Using the class cCPUTicker the actual time is found out and as soon as the time determined by the sampling period elapses, the whole cycle for computing the control signal is repeated.
- *Visualization of results to the local user.* The user can receive numerical information about the actual position of the pendulum and its angle. Moreover, there is also visualized a simplified version of the animation that is available to the remote user, too.
- *Communication with the remote client.* Actually, the communication is realized by 2 threads. The task of the first one is to wait for the request of the first remote user, to process this request and to give him or her back the asked data. Meanwhile, the second thread informs all next interested remote clients that the experiment is running and they have to try to connect later. As soon as the experiment ends the first thread is again prepared to serve to the first interested remote user.

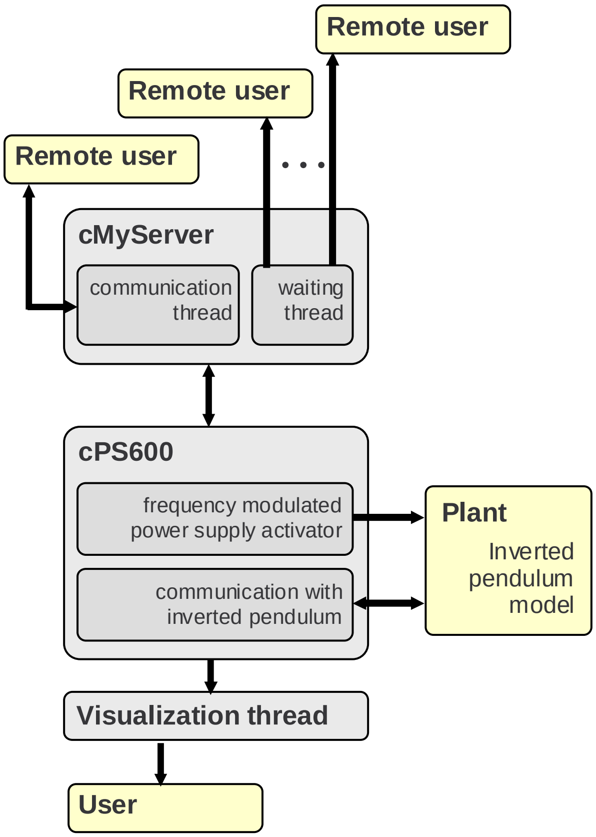
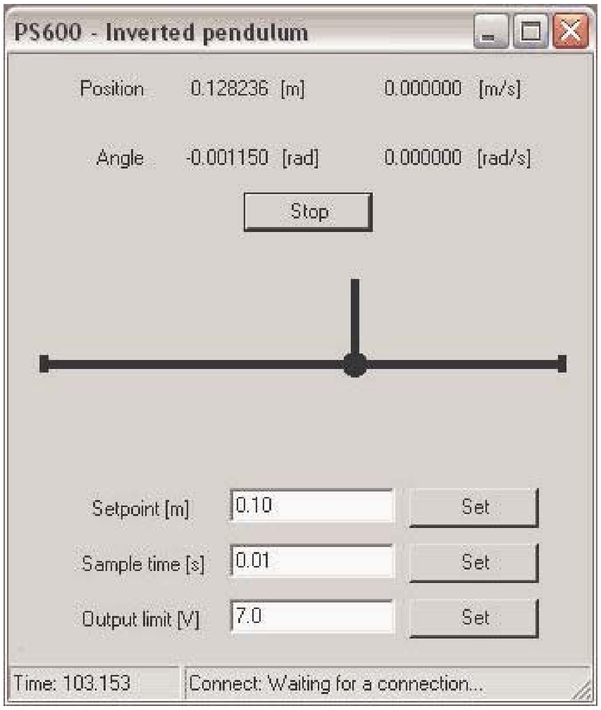


Fig. 3. Overview of threads in the server application

The first step to start the remote experiment is to execute the server application. Its activating causes that the equipment is synchronized. It means it is initialized into the initial zero position that is situated in the middle of the pendulum path. The processor tick counter is also synchronized with the processor.

The server application (Fig.4) presents a simple local graphical user interface that enables administrator to set and verify most of activities that will be later available to the remote user.



PS600 - Inverted pendulum

Position 0.128236 [m] 0.000000 [m/s]

Angle -0.001150 [rad] 0.000000 [rad/s]

Setpoint [m] 0.10

Sample time [s] 0.01

Output limit [V] 7.0

Time: 103.153

Connect: Waiting for a connection...

Fig. 4. The server application

One can find here the numerical information about the position of the pendulum and the angle. In the second column there are also numerical values of derivatives of both variables. Then one can find here the button for starting and stopping the plant control, the sketch animation and finally the area for the setting of parameters. The application seems to be very simple. However, their main task is to enable to client to connect to the inverted pendulum remotely.

B. *Client Side*

As it was already told the client application was prepared using Macromedia Flash. For the communication with server it is needed to create a new instance of the native class XMLSocket. The communication is ensured by the corresponding methods (Fig.5).

The method connect() serves for connecting to the server. Its arguments are the server IP and the number of the port that is used for communication.

The method onData() enables to receive and process data from the server. The data are in the XML form and therefore they have to be separated to single values using a XML parser. Only after the parsing data can be used for numerical and graphical visualization of results.

The send() method is used for sending all requests to the server. It is needed to set the pendulum required position or the sampling period and also to switch on and off the whole experiment.

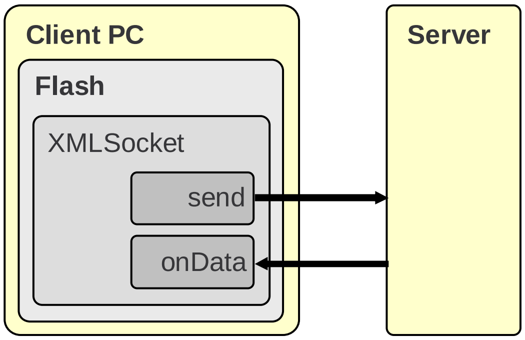
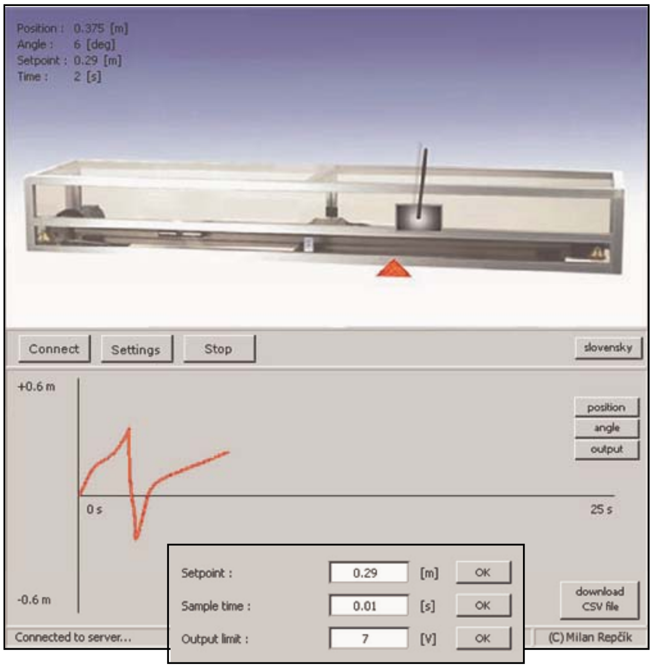


Fig. 5. Client-server communication

The final client application (Fig.6) enables to visualize results. The main attention is devoted to the model animation. User can also follow numerical values of the required position of the pendulum, the actual position of the pendulum, its angle and the elapsed time. The position is given in meters whereby the zero position is in the middle of the pendulum path. In the right direction the position is determined by positive values and in the left direction by negative values. The angle in the top position is 0º and in the bottom position ±180º.



Position : 0.375 [m]
Angle : 6 [deg]
Setpoint : 0.29 [m]
Time : 2 [s]

Setpoint : 0.29 [m]
Sample time : 0.01 [s]
Output limit : 7 [V]

Connected to server...

(C) Milan Repčík

Fig. 6. The client application

In the middle of the screen one can find menu for operating the whole application. The user connects to or disconnects from the server by the first button. The second button enables to change the bottom part of the window that can be switched between the setting and graph mode. The last button gives the user a possibility to start and to stop the control of the plant provided by the predefined controller.

As it was already mentioned the bottom part of the window could be changed to two modes: the setting mode and the graph mode. In the setting mode users determines parameters for the remote control. He or she can numerically define the required value of the pendulum position, the sampling period and the saturation of the control variable.

The graph mode shows the graphical dependencies of all followed variables (position, angle, control variable). The user can choose what he or she would like to see by the clicking on the corresponding name of the variable. The measured data can also be saved into the local file and visualized later in some external software environment, e.g. in Matlab, Excel, etc.

The functionality of the flash animation is realized by the switching of two frames that have together 3 layers. In the first layer there is always displayed the animated plant and the menu control buttons. This layer occurs in both frames. Elements for the setting mode are placed in the second layer and elements for the graph mode are in the third layer. The second and the third layers are given only into one frame whereby these frames are switched by the middle control button (Set).

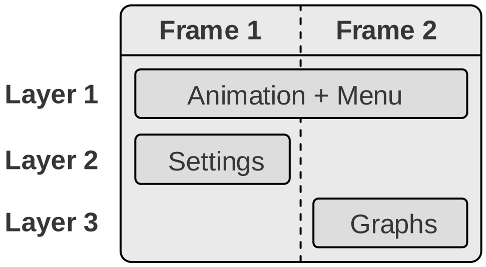


Fig. 7. Frames in the Flash client application

VI. Conclusion

Many approaches to remote control use the engine of very popular Matlab simulation software package. It facilitates the whole design procedure because it is necessary to take care only about the communication via Internet and not about computations and tasks that could be accomplished by Matlab, e.g. the numerical methods needed for simulations.

In the paper we presented another way of realization. This type of remote control enables to skip all intermediate stages between the application and the operating system that can limit fluent running of the application and its velocity. In addition, the used multimedial format Flash has only low requirements on the velocity of Internet connection. Therefore the introduced solution can be a good alternative for remote experiments.

Acknowledgment

Author thanks to Milan Repčík and Michal Sedlák for valuable assistance and many discussions on the topic.

References

[1] P. Bisták, “Remote Control of Thermal Plant Using Easy Java Simulation”, in *Int. Conf. on Interactive Computer Aided Learning ICL’06*, Villach, Austria, 2006.

[2] P. Bisták, K. Žáková, “Organising Tele-Experiments for Control Education”, *11th Mediterranean Conference on Control and Automation*, Rhodos, Greece, 18.-20.6.2003.

[3] M. Huba, P. Bisták, M. Fikar, M. Kamenský, “Blended Learning Course ‘Constrained PID Control’”, *7th IFAC Symposium on Advances in Control Education ACE’06*, Madrid, Spain, 2006.

[4] M. Huba, M. Kamenský, “Constrained Magnetic Levitation Control”, *16th IFAC World Congress*, July, Prague, Czech Republic, 2005.

[5] P. Píš, K. Žáková, “Remote Control of the Beam and Ball Model,” *Process Control 2005*, Štrbské Pleso, Slovakia, June 2005, pp.201.1–201.6.

[6] M. Repčík, K. Žáková, „Remote Control of Inverted Pendulum“, *Remote Engineering & Virtual Instrumentation : International Conference REV*, Porto, Portugal, 2007.

[7] Chr. Schmid, “Virtual Laboratory for Engineering Education”, *ICDE conf.*, Vienna, Austria, 1999.

[8] Chr. Schmid, „Internet - basiertes Lernen“, *Automatisierungstechnik*, 51, No. 11, pp. 485-493, 2003.

[9] M. Šimunek, P. Bisták, M. Huba, “Virtual Laboratory for Control of Real Systems”, *Conference Procedings ICETA*, September, Košice, Slovakia, 2005.

[10] K. Žáková, M. Huba, V. Zemánek, M. Kabát, “Experiments in Control Education”, *IFAC Symposium on Advances in Control Education*, December, Gold Coast, Australia, 2000.

[11] K. Žáková, M. Sedlák, Remote Control of Experiments via Matlab, *Int. Journal of Online Engineering (iJOE)*, Vol. 2, No. 3, 2006.

[12] JMatLink Version 1.00;

[13] http://www.held-mueller.de/JMatLink

[14] http://www.learnframe.com

[15] http://www.amira.de

[16] http://www.answers.com.

Katarıına Žáková graduated in Technical Cybernetics from the Slovak University of Technology, Slovakia in 1991. She obtained her PhD. degree in Automation and Control from the same university in 2003. Since 1991 she is with the Department of Automation and Control, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology. Her current research activities are connected with remote control of plants, synthesis of nonlinear systems, computer aided control design, computer algebra systems and e-learning.