Remote control of inverted pendulum

Milan Repčík and Katarína Žáková
Slovak University of Technology in Bratislava
Faculty of Electrical Engineering & Information Technology
Bratislava, Slovakia

Abstract—The inverted pendulum is a classical problem in control theory and widely used as benchmark for testing control algorithms (PID controllers, neural networks, Genetic algorithms, etc).

In the paper one can find the description of the remote control for the presented plant. We developed own server (C++) and client application (Flash). The user can run the experiment remotely and can receive results in the graphical form and in the form of a simple animation. The gained experience is also discussed in the paper.

Index Terms—remote control, inverted pendulum, server and client application.

I. INTRODUCTION

Technology is increasingly changing our lives. First, it was the emergence of the desktop computer, now it is the Internet. It enables the ordinary person to have access to never-ending quantities of information and knowledge. As more individuals become connected, the Internet will penetrate deeper into our everyday activities, including the way we learn. The term e-learning covers a wide set of applications and processes, including computer-based learning, Web-based learning (online Learning), virtual classrooms, and digital collaboration. E-learning can be defined as the delivery of content via all electronic media, including the Internet, intranets, extranets, satellite broadcast, audio/video tape, interactive TV, and CD-ROM. Online learning constitutes just one part of technology-based learning and describes learning via Internet, intranet, and extranet [10].

In engineering education providing hardware and software components of existing laboratories via the Internet creates a base for establishment of virtual laboratories. 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.

Design of control applications that are available via Internet is oriented in two directions: control of virtual devices and control of real physical plants. The first possibility enables to simulate the virtual model on a computer and with Internet access to offer it as an animation to students via the WWW or CD-ROM. However, using the animation models cannot substitute the work with real physical plants that always demonstrate some unmodelled dynamics, parasitic noise, friction, etc. Unfortunately, the number of students is high in comparison with the number of available real plants. A possible solution of this problem is building of remote lab that gives learners access to laboratories via Internet.

For accomplishing the remote control of experiment it is necessary to create two applications (Fig.1): application for the server and application for the client.

The paper demonstrates one way how to use remote control for the inverted pendulum model. The inverted pendulum is one of the most important 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. PLANT

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.2) is represented by a revolving pendulum mounted on top of a moving base [11]. 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.

III. STARTING POINTS

A. Server Side

The choice of programming tools for the preparation of the server side of experiment was 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** – the effort was to skip all intermediate stages between application and operating system.
- **Accessible hardware and software** – it was necessary to take into account the most used systems and software environments at the universities and their accessibility for users.

The first requirement can be fulfilled by the selection of the programming language with good support of the object-oriented programming. One can choose from languages such as .NET, Basic, Delphi, C++, Java, Python, and JavaScript. Matlab with its toolboxes also offers an alternative to the design of remote control of plants [9].

The second requirement excludes the JavaScript and similar scripting languages since these languages are only the presentation languages. They don’t send commands to the operating system directly and therefore they need an additional tool for communication between application and operating system. The Java language brings the similar situation. In spite of the fact that Java is a multi-platform instrument it also needs an extra tool for communication with application. It can be easily understood that such indirect connection between the operating system and application limits the fluent running of the application and its velocity. Simply said, such applications are slower.

Finally, at the choice of the programming tool we had to realize that the most of computers at the university is equipped by the operating system Microsoft Windows and therefore the design should be oriented in this way.

Summarizing these facts 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.

B. Client Side

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** – it should guarantee smooth animations.
- **enable simple Internet communication** – the communication can be implemented e.g. using sockets.

- **create a multi-platform application** – the platform independence is important because the application is presented to users via Internet.

The introduced requirements can be fulfilled by several solutions. Let’s introduce some of them. One can use:

- **combination of JavaScript and CSS**. In this case the graphical layout of the HTML web site is ensured by the Cascading Style Sheets that can be dynamically changed by JavaScript. The disadvantage is that the animation is not very smooth and there is relatively complicated communication with the server.
- **Java applet**. Since Java enables a comfortable network programming, the communication is very simple. However, there still exist the problem with smooth animation without a blinking. This problem can be solved, but it is uselessly tricky and complex.
- **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.

We decided to use the third mentioned possibility.

IV. REALISATION

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.

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 algorithms 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 that is 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 enables parallel processing of several tasks. This is enabled by using several threads (Fig.4).

![Figure 4. Overview of threads in the server application](image)

Firstly, it is necessary to activate 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.

Next, one has to detect the actual state of remote experiment. 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 elapsed the whole cycle for computing the control signal is repeated.

The next thread is responsible for visualization of results to the possible local user. He or she 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.

Further, there are 2 threads that ensure communication with the remote client realized by means of the Flash animation. 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.

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 is shown in Fig.5. It 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.

The server application is shown in Fig.5. It 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.

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.6).

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. Firstly, ActionScript processes the request for actual results from the remote plant 25 times per 1 second. Then, the second type of requests is related to the setting of the remote plant. It is needed to set the pendulum required position or the sampling period and also to switch on and off the whole experiment.

The final client application can be seen in Fig.7. The upper part serves for visualization of results where the main attention is devoted to the model animation. By the red triangle the required position of the inverted pendulum can be determined. In the left top corner the 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.

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 (Fig.8) 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).

On the bottom margin of the window is situated a status bar. The user can find here information about the connection to the server (“Waiting for connection”, “Connected to server”, “Connection closed”) or about the server status (“Server is offline”, “Server is busy”).

The flash client application is prepared in two languages mutations – English and Slovak.

V. CONCLUSIONS

The paper presents one way of the remote experiment realization. Its main contribution consists in an atypical way of the solution. The client application is designed using the widespread multimedia format Flash. It enabled to create a simple and effective animation that has only low requirements for the velocity of Internet connection. The server application working on the Microsoft Windows platform can also be easily used without a deep knowledge of the programming code. From this reasons the presented solution can be a good alternative to other techniques that enable to accomplish experiments in remote laboratories.

ACKNOWLEDGMENT

The work has been supported by ESF Project JPD BA3, No. 13120120183 and by Slovak Grant Agency, Grant KEGA No. 3/3121/05 and Grant VEGA 1/3089/06. This support is very gratefully acknowledged.

REFERENCES


AUTHORS

M. Repčík, K. Žáková are 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).