

Two Ways of Inverted Pendulum Remote Control

KATARÍNA ŽÁKOVÁ

Faculty of Electrical Engineering and Information Technology
Slovak University of Technology
Ilkovičova 3, 812 19 Bratislava
SLOVAKIA

Abstract: - The paper is focused on remote control of a typical nonlinear system that is used as a benchmark in control education at many universities. Two various approaches of the problem solution are presented. In the first one there was created the Java client-server application that enables to connect the graphical user interface with the Matlab engine running on the server. The second approach presents a client-server application that connects .NET technology and Macromedia Flash format.

Key-Words: - Inverted Pendulum, Remote Control, Server Application, Client Application, Java, C++, Flash

1 Introduction

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.

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 process of interactive learning through testing, evaluation, decision-making, and error correction creates a much stronger learning environment than passive instructions.

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.

The paper demonstrates two ways 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.

2 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.

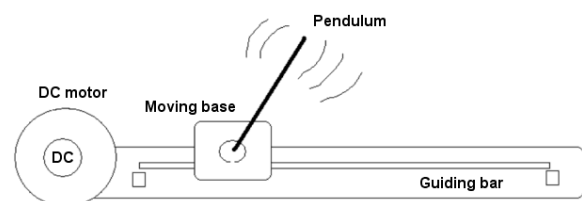


Fig. 1. Sketch model of inverted pendulum

The plant (Fig.1) is represented by a revolving pendulum mounted on top of a moving base [9]. 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.

3 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 two sections two different solutions for remote control of experiments are presented. We tried to use various software tools. It brings different possibilities and an advantage of later comparison.

4 Remote Control via Matlab Engine

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. Actually, as it was mentioned already before, it was necessary to prepare both the server and the client applications.

4.1 Server Side

The server is an application that listens at a particular port and responds to clients. It could be

any of standard servers like http, ftp or a custom server made for our specific application.

Another possibility is to create a Java proxy server interpreting user commands from the client application to Matlab. For this purpose it is possible to use a dll library JmatLink [7] that accesses services and data of Matlab. The library is implemented to the simple class JmatLink.class that can be used inside of created Java proxy server.

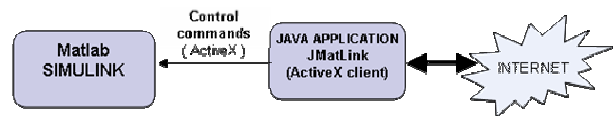


Fig. 2. JMatLink control channel

Proxy server can be realized as a Java application with the structure shown in Fig.2. On the one side there is a TCP/IP server that receives command from the client (Java applet). These commands are transformed to the Matlab commands and sent to the Matlab workspace using methods of JMatLink library.

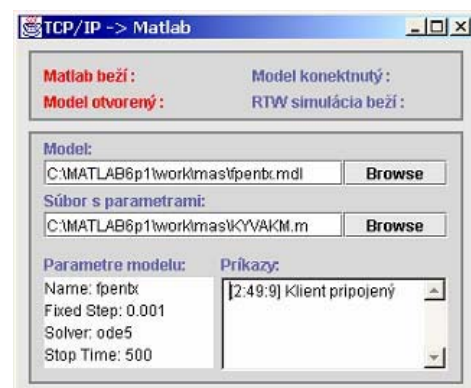


Fig. 3. Java server application

Proxy server has 3 basic functions:

- to start the real time simulation,
- to stop the real time simulation and
- to change the model parameters during the simulation running.

4.2 Client Side

In general, TCP/IP client is an application that connects to a specific port on a TCP/IP server and exchanges data either as a stream or text.

It is installed on a web server that can be approached by a student via Internet. It is very often realized as Java applet that enables to modify controller or simulation parameters and visualize simulation results. In Fig.4 the applet for an inverted pendulum control is shown. After its opening in a

user web browser, TCP/IP client connects to the proxy server and to the S-function in the Simulink model. An user is informed about the time of connection and the connection status through short information note in the first part of the applet window („client is connected“). In this moment, everything for real-time simulation is prepared.

The user can enter a required position of the pendulum. The buttons „Start“ and „Stop“ serve for the control of simulation running. During the simulation user can follow numerical values of the carriage position, deviation angle of the pendulum, time of simulation and the graphical dependence of the carriage position. The simulation results can also be visualized through model animation that can run in two modes: on-line and off-line mode.

In the on-line mode the animation starts immediately after the beginning of simulation and in each moment user can see actual state of the pendulum in the animated model. The time between the real time experiment and the animation is synchronized. However, the velocity of this animation depends on the velocity of data transfer via Internet. If the transfer is not sufficiently fast, the animation is not capable to visualize the samples from experiment on line because of big delays. Then, the off-line animation can be used. In that case the real time simulation is realized only numerically without the contemporaneous animation. After the simulation ends, measured data are sent to the client and the animation can be accomplished.

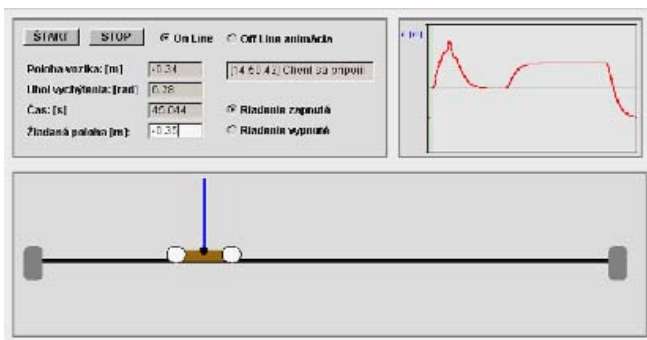


Fig. 4. Java client application

So, it means the Java client has to enable to user

- to control the process and
- to collect the data from the process

Connection between Matlab and Java client can be realized directly using TCP/IP communication. TCP/IP server can be realized by Matlab S-function. It enables transfer of data from Matlab to the connected TCP/IP client (Java applet). The Matlab

S-function operating as TCP/IP server is placed in Matlab/Simulink model whereby the input of this block is formed by signals that should be transferred to the client.

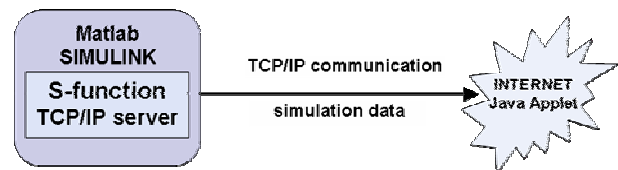


Fig. 5. TCP/IP server on Matlab side

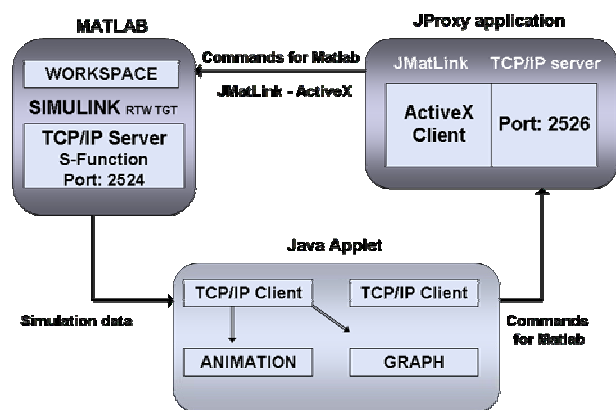


Fig. 6. Remote control of the plant via Internet

5 Direct Control of Remote Plant

In the second approach we tried not to use Matlab engine since actually it is not available everywhere. Since it was necessary to prepare everything from the beginning we decided to change the programming language, too.

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.

5.1 Server Side

The server application enables parallel processing of several tasks. This is enabled by using several threads (Fig.7).

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 $\pm 10V$. Since the used DC motor can work with much better performance that the card can supply the provided signal has to be amplified.

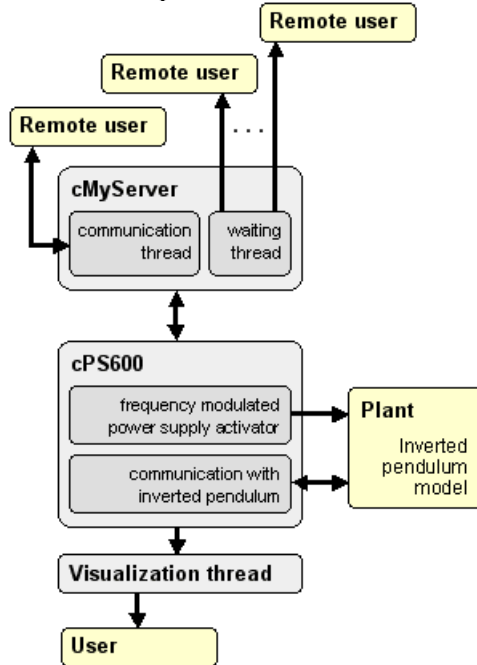


Fig. 7. Overview of threads in the server application

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 `cPUTicker` that computes the processor ticks, the actual time can be found out. 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 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.8. 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.

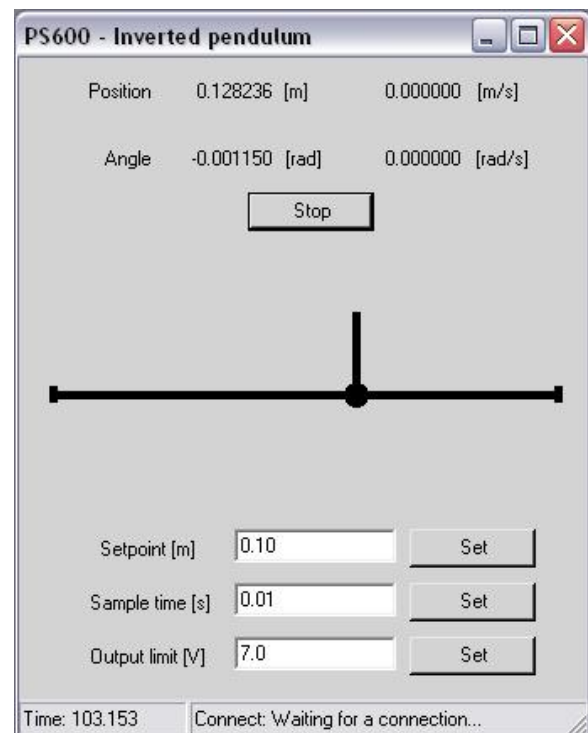


Fig. 8. The server application

5.2 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`. Firstly, user has to connect to the server. For this purpose he or she has to specify the server IP and the number of the port that is used for communication. Then, the client-server communication is ensured by the following two methods.

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. ActionScript processes the request for actual results from the remote plant and 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.

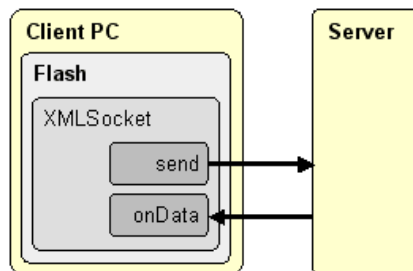


Fig. 9. Client-server communication

The final client application can be seen in Fig.10. The upper part serves for visualization of results where the main attention is devoted to the model animation. 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.

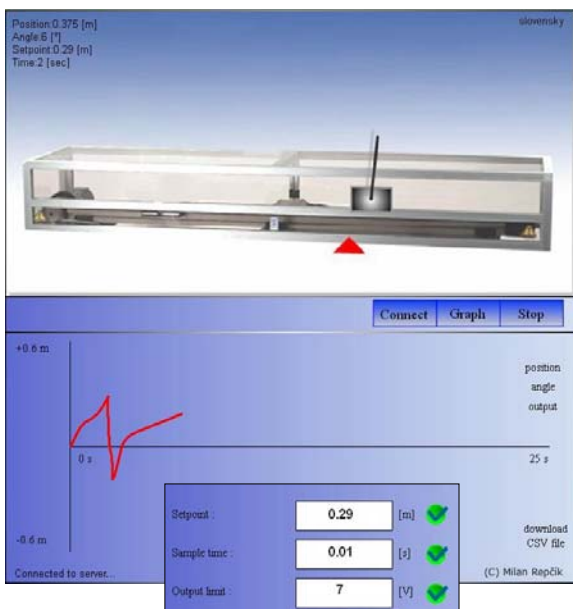


Fig. 10. The client application

In the middle of the screen one can find a simple 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 required position of the inverted pendulum can also be set by the red triangle directly in the animation.

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.11) that have together 3 layers. In the first layer there is always displayed the animated plant and the menu with 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 the user can find 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").

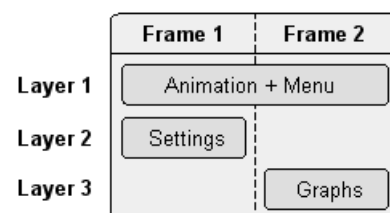


Fig. 11. Frames in the Flash client application

6 Conclusion

In the paper we presented two various ways for remote control of experiments

The first approach used the engine of very popular Matlab simulation software package. It facilitated the whole design procedure because we

had 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 spite of this the second approach enabled to skip all intermediate stages between the application and the operating system. They can limit fluent running of the application and its velocity. In addition, the used multimedial format Flash has also only low requirements on the velocity of Internet connection.

References:

- [1] P. Bisták, "Remote Control of Thermal Plant Using Easy Java Simulation", *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] P. Piš, K. Žáková, "Remote Control of the Beam and Ball Model," *Process Control 2005*, Štrbské Pleso, Slovakia, June 2005, pp.201.1–201.6.
- [4] Chr. Schmid, „Internet - basiertes Lernen“, *Automatisierungstechnik*, 51, No. 11, pp. 485-493, 2003.
- [5] M. Šimunek, P. Bisták, M. Huba, "Virtual Laboratory for Control of Real Systems", *Conference Proceedings ICETA*, September, Košice, Slovakia, 2005.
- [6] K. Žáková, M. Huba, V. Zemánek, M. Kabát, "Experiments in Control Education", *IFAC Symposium on Advances in Control Education*, December, Gold Coast, Australia, 2000.
- [7] JMatLink Version 1.00;
<http://www.held-mueller.de/JMatLink>
- [8] <http://www.learnframe.com>
- [9] <http://www.amira.de>