

A VIRTUAL LABORATORY FOR AN INVERTED PENDULUM AND CRANE CONTROL

I. Masár, C. Röhrig, A. Bischoff, M. Gerke, H. Hoyer

*Department of Electrical Engineering
Control Systems Engineering Group
FernUniversität in Hagen
Universitätsstr. 27, 580 97 Hagen, Germany
Ivan.Masar@FernUni-Hagen.de*

Abstract: A virtual laboratory allows students to gather experiences with practical on-line experiments at remote sites. Moreover, it becomes possible to share expensive equipment among several universities or education centres. Some German universities were engaged during the last three years in the 'LearnNet' project, in order to develop a network of remotely accessible virtual laboratories and additional courses in control and automation. One of our contributions to this project is a laboratory experiment for the inverted pendulum and gantry crane control. Several new components for real-time control and Internet access have been developed and implemented. *Copyright © 2004 IFAC*

Key words: Virtual laboratory, inverted pendulum/gantry crane, xPC Target

1. INTRODUCTION

In the past few years 'Virtual labs' have been meaningful and very popular for tutorials in control theory, since they allow performing of interesting experiments dedicated to real-time control of technical systems all over the world. At our research group, several such laboratories were established, including a lab experiment for remote control of a mobile robot with omnidirectional wheels (Röhrig and Jochheim, 2000; Bischoff and Röhrig, 2001) and for Internet-based testing and programming of industrial controllers (Simatic S7 family, Siemens). At the present time, we developed a new virtual laboratory for real-time control of an inverted pendulum and gantry crane system. This lab is one of our contributions to the 'LearnNet' project, in which several German universities collaborate in order to set up a common platform for on-line learning and experimentation in control theory.

2. VIRTUAL LABORATORY OVERVIEW

As mentioned above, our new virtual lab allows a teleoperated control of the inverted pendulum and

gantry crane system, respectively. This system was chosen because it represents one of the most commonly used non-linear systems in control theory at the undergraduate level. The ability of this system to provide interesting and challenging experiments makes it very attractive for students. It serves as a remote experimental environment to exploit various control techniques and thereby to provide comparative study. At this time, the students can explore several control algorithms for various system configurations, including state-feedback, PID and fuzzy controllers.

For realistic presentation of the experiment via the Internet, several visualization methods are used. The students choose from video- and audiostreams, 3D animated graphics or conventional curve plotting, depending on the speed of their data connection to our lab. Only standard Web-browsers (optional with VRML Plug-In and Java Media Framework) are required as an interface to the laboratory.

Main components of our virtual laboratory and their interconnections are depicted in Fig. 1.

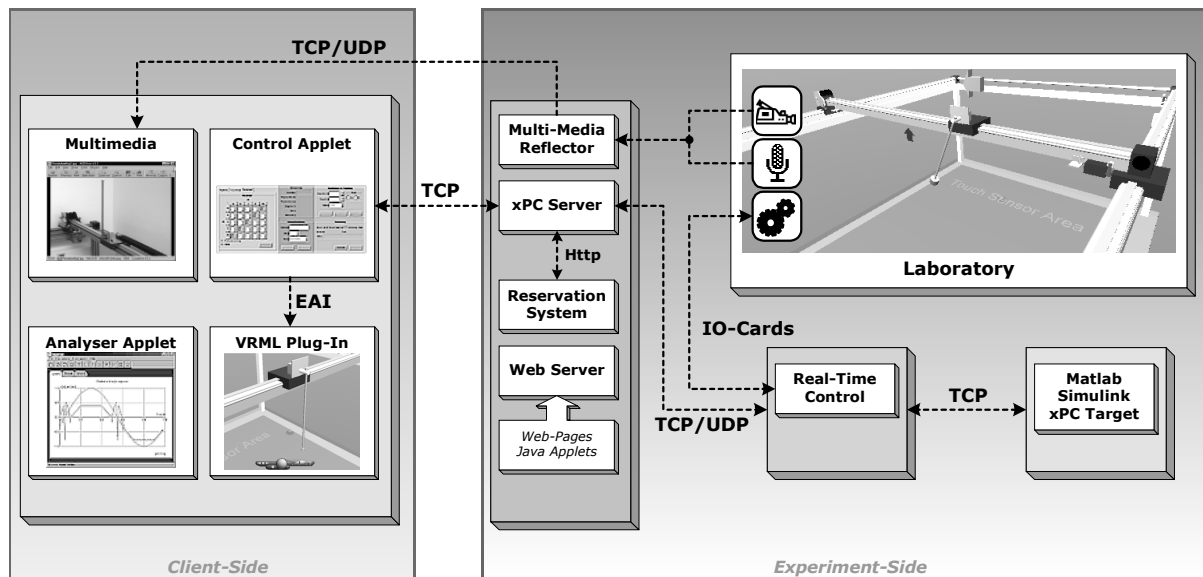


Fig. 1. Virtual laboratory components

Software components and tools used in the lab can be organized into three groups:

- components for experiment set-up (control algorithms design, procedures set-up, real-time control of the pendulum/crane system) – Matlab/Simulink/xPC Target,
- server components interconnecting clients with the experiment – xPC Server communicating with the reservation system, Multimedia Reflector, Web Server,
- client components for remote lab operation and experiment visualization – a Control Applet, Java Media Framework, an Analyser Applet, and a VRML Plug-In.

3. EXPERIMENT SET-UP

The essential part of the laboratory, the inverted pendulum/gantry crane system, is shown in detail in Fig. 2. It is a triaxial system, which consists of the gantry movable along an x-axis and pendulum/crane system also movable along y-axis placed on it. The DC motors drive both the gantry and the cart. For control purpose, a standard PC with input/output cards is used. These cards convert the signals from the sensors (optical encoders for measurement of the axes positions; contactless proximity sensors) and control the drives for the axes.

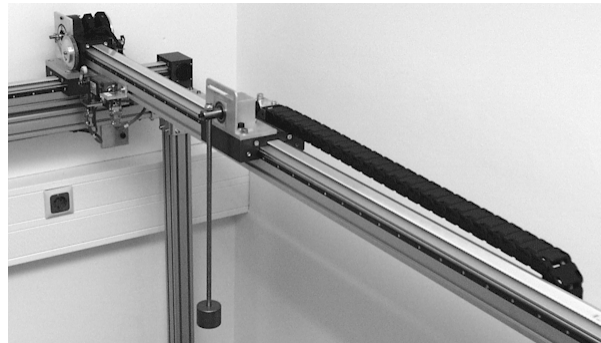


Fig. 2. Inverted pendulum/gantry crane system

The control algorithms were designed and implemented by the Matlab/Simulink software package (Release 12) with Real-Time Workshop and xPC Target Toolbox. These tools are able to automatically generate stand-alone real-time applications from the Simulink models that run on the so-called *Target PC*, while their development is carried out on separate host computer (*Host PC*). The Target PC is based on a special real-time operating system (Real-Time Kernel) and it communicates with the Host PC through serial link or network connection. Through this connection it is possible to operate the Target PC (load/start/stop application, change algorithm parameters, trace signals) completely by the Host PC (from the Matlab command line or directly from the Simulink model). The major advantages of this experimentation platform are:

- short development times for the control algorithms,
- only standard PC-hardware requested for the Target PC (486/8MB PC is often suitable minimum, we used a Pentium MMX/200MHz/32MB),

- high stability and reliability of the Target PC, since it works with an optimised operating system (reduces to necessary functionality).

4. SERVER COMPONENTS

Server components control the Internet access to the lab and the course of the experiment (xPC Server with reservation system) and exchange the data between the user and the lab.

4.1 xPC Server

Even though the Target PC is fully controllable from the Host PC, this feature has been used only in the development and testing phase of the lab. For the teleoperation of the lab, some type of remote control of the Target PC via Internet is necessary. With the xPC Target Toolbox, a simple interface to the xPC Target through Web-browser is provided too. Since it was not possible to implement all required functionality with this interface, we decided to use C-API functions from the supplied library *xpcapi.dll* to build our own server application in order to manage the Target PC, called *xPC Server*. These functions serve to establish a basic control for the Target PC (restart, opening/closing of the communication port), downloading of any real-time applications and control of its functional flow (load/unload, start/stop, stop time and sample time setting), adjustment of the control algorithms and signal tracing. Moreover, the xPC Server communicates with the client components, enables the control of the lab cameras and supervises the access to the lab.

One of the main advantages of the realized xPC Server is its easy adaptability to new experiments and other lab equipment. Namely, each experiment is implemented by a Simulink model. This model contains the particular control algorithm as well as some special blocks that are required for operation with the xPC Server. These blocks are build for:

- the initialization of the experimental plant,
- turning on/off of the control algorithm,
- acquisition and transmission of the experimental data,
- camera control.

However, all these requirements are easy to accomplish, as the necessary blocks and subsystems are for the disposal in advance.

A further condition for the compatibility with the xPC Server, is the initialization file created for a specific Simulink model. This file includes information about the model, the list of the adjustable parameters with their boundary values, the numbers of the captured signals, etc. The xPC Server reads this file before loading the model on the xPC Target and sets the parameters of the communication with the client accordingly.

In the xPC Server, two techniques for data logging are implemented. For data acquisition during the execution of the lab experiment, the fast (but unreliable) UDP protocol is used. This type of communication is supported directly by the xPC Target Toolbox. Therefore, only dedicated blocks must be added into the Simulink model with the tested algorithm. The xPC Target sends data during running of the application through its UDP port to the xPC Server, which transfers them further to the client. In this manner received data are used on the client-side for the animation of the 3D-model and on-line monitoring tools.

For any more precise analysis of the experiment, another method for data storage is used. Since the data for analysis must be uniformly sampled, we used special blocks from the xPC Target Toolbox in Simulink models – *xPC Scopes*. These data are logged for a desired time on the xPC Target first and then they are sent to xPC Server all at once. On the server side, these data are saved in files, which are available for download by the student.

The last function of the xPC Server is the remote operation of the cameras in the lab. In our lab, several cameras are placed that present a view on the experiment from various viewing angles. The user can teleoperate the cameras (rotating and zooming) and thereby adjust them according to his needs. Besides the videostream from the cameras, the xPC Target screen is transmitted to the client, who can thereby observe its operating condition.

4.2 Multimedia Reflector

The *Multimedia Reflector* (MMR) is a server component developed during the LearNet project (Ruhr-Universität Bochum). It is designed for real-time processing of video-, audio- and data-streams from the laboratory and for their live transmission (synchronously) to the client. MMR simultaneously streams to several clients with various transfer rates.

In addition to remote experimentation via Internet, MMR can be used during any conventional training, too. Here the students can observe the experiment on several workstations (e.g. for large groups, for small experimental set-up's). The MMR server is based on Java Media Framework (JMF) functions provided by Sun.

4.3 Reservation System

A new reservation system for distributed labs has been developed during this project and it is now used for booking of time quota with respect to each experiment in charge. This system is accessible through Web-browser interface and offers information about occupation of particular

experiment. The sample reservation interface is presented in Fig. 3.

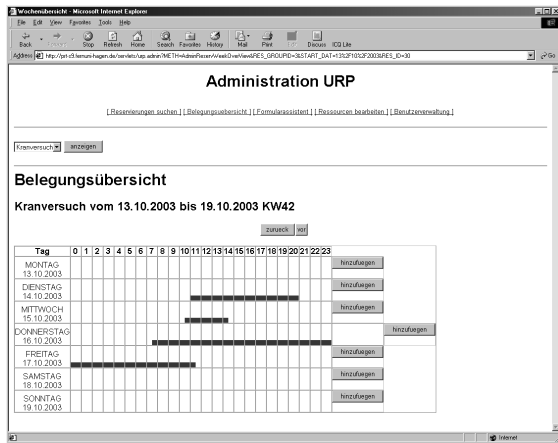


Fig. 3. Interface to the reservation system

Before accessing any lab experiment, every user has to identify by his user name and password that has been distributed to him by the reservation system. These data are checked by the xPC Server, which communicates with the reservation system server via HTTP protocol. The response of the reservation system is an XML-document with allowed duration time for teleoperation of the experiment and acknowledgement of the access to the lab. The xPC Server executes this checkout in regular time intervals during experimentation in order to assure the access to the laboratory for the next client.

5. CLIENT COMPONENTS

Client components are dedicated to experimental flow control and data visualisation. The communication with all servers involved is based on the TCP protocol and can be handled by standard Web-Browsers. All client components are implemented as JAVA-applets. Thereby, a comfortable and easy to operate user interface is created. The main requirement on the clients side can thus be reduced to standard Web-browser without additional software.

Because of several types of available web-browsers and varied transfer rate of the Internet connection, a number of different equipped interfaces to the lab (full version for the students with the fast connection, the version with only VRML-model animation for the clients with slow connection, etc.) were designed. The Web-interface to the gantry crane controlled by a fuzzy controller is shown in Fig. 4.

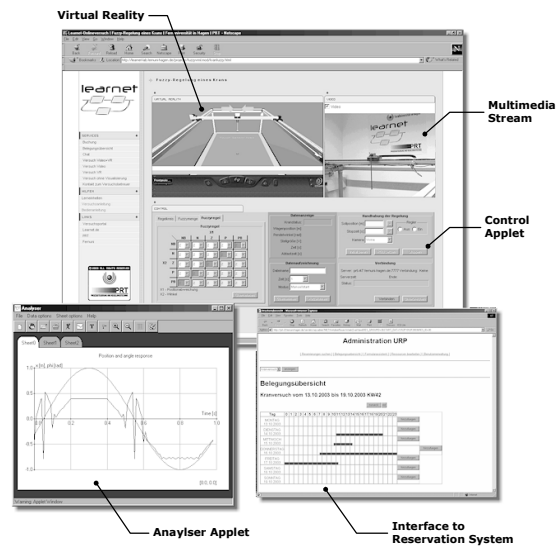


Fig. 4. Web-browser with user interface to the lab

5.1 Control Applet

The main component for teleoperation of the virtual lab is the *Control Applet*. A version for gantry crane control is shown in Fig. 5. The tasks performed by the Control Applet are:

- Verification of the access to the lab
- Opening/closing connection with the xPC Server
- Transmitting the commands to the server in defined protocol (start/stop application, get/set parameters, etc.)
- Displaying experiment state (elapsed time, server response, etc.) and signal values
- Controlling of the communication status and user activity (when no activity occurs, it aborts the experiment and closes the communication)
- Communication with any VRML Plug-In

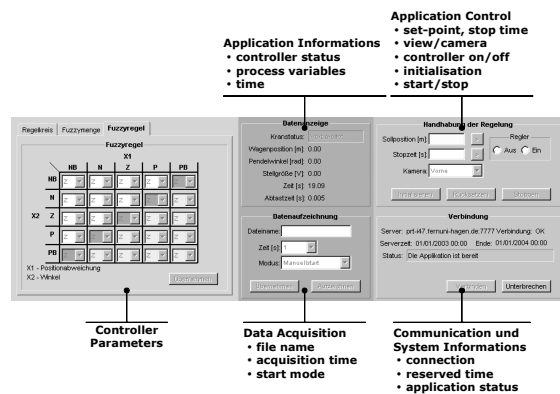


Fig. 5. Control Applet

5.2 VRML Plug-In

A VRML Plug-In is used for visualization of the experiment by means of its virtual 3D model. This

option is used because it allows realistic presentation of the lab scenery in virtual reality. Moreover, it works even with low-bandwidth Internet connections, because of the small amount of transmitted data. Furthermore, the virtual reality model enables to visualise miscellaneous physical and process quantities like acting forces, set-points, control signals, etc., and to interactively handle the lab by various switches and sliders. The animation of the virtual scene is implemented by using the External Authoring Interface (EAI).

The 3D model of the pendulum/crane system is shown in Fig. 6. In this model, the user can change the desired position of the crane by so-called touch sensor.

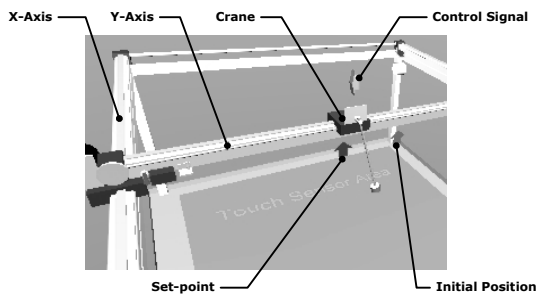


Fig. 6. VRML 3D model of the pendulum/gantry crane system

5.3 Analyser Applet

A special 'Analyser Applet' has been developed for the on-line experiment evaluation; it is useable for plotting of the experimental data saved on the server. The data are received by the 'Control Applet' from the xPC Server and saved in the user's local file system. The user can plot several curves, zoom-in/out the graph, etc. This applet is shown in Fig. 7.

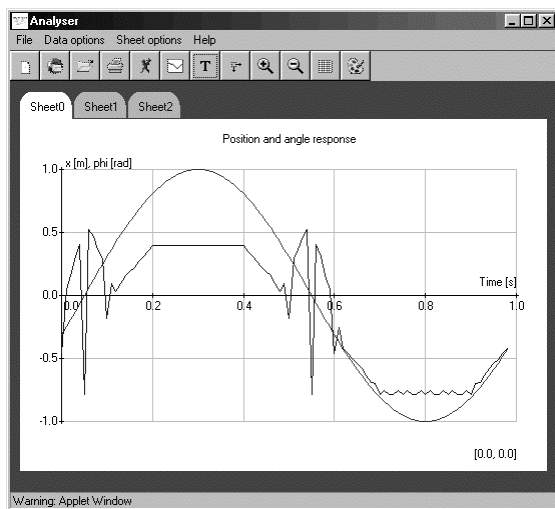


Fig. 7. Analyser Applet

5.4 Multimedia Applet

An additional 'Multimedia Applet' (Ruhr-Uni-Bochum: in combination with MMR server) displays streaming video from the virtual lab, plays sound and roughly plots the process data. It is a hybrid Java-class that can be started as Java applet or as stand-alone application. It exploits JMF, which must be installed on the client computer. On the start-up, the user can choose from the set of available media (video, sound, data) as well as transfer rate for the video stream. This 'Multimedia Applet' receives the data directly from the MMR server. Examples of video streams from several cameras located in our lab as well as transmitted xPC Target screen are given in Fig. 8.

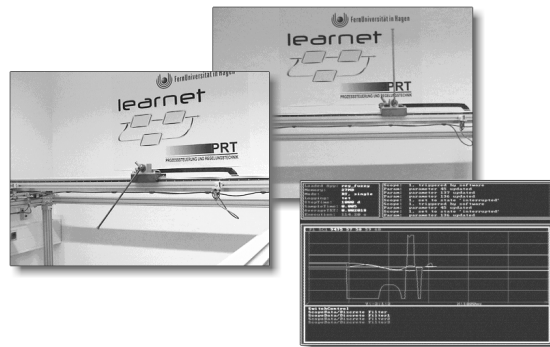


Fig. 8. Video streams from the lab

6. LABORATORY UTILIZATION

As the FernUniversität in Hagen is the only distance teaching university in Germany, such virtual labs play significant role in its educational concept, particularly in the engineering-oriented fields of study.

The presented laboratory provides a wide range of executable experiments with various complexity and it can be used as the test platform for verification of many types of the control strategies. The spectrum of the experiments ranges from the position/speed control of the single axis to the control of the pendulum (alternatively 2D-pendulum) in its stable/unstable position by several types of controllers (PID, state-feedback, fuzzy controller, etc.).

The laboratory was completed during the summer term of the last year and the first students were testing the lab also in the last summer and winter term. The tests were performed through both LAN and modem connection with satisfactory results. At the present time, we offer the exercise in the gantry crane configuration as the part of several courses from the control theory and mechatronics.

CONCLUSION

This paper describes the concept and implementation of our new 'virtual laboratory' for the inverted pendulum/gantry crane control.

The components used in the lab as well as on the client side have been discussed here. A novel architecture of the virtual laboratory based on utilization of the xPC Target Toolbox from Mathworks has been designed and is outlined here in detail. The presented structure of the lab together with all its components provides a possible framework for future labs.

The technical structure of the lab allows further addition of new control algorithms/problems to solve and thereby vary the education process for the students.

First experimentation with our remote lab carried out by our students and an additional evaluation process proved the functionality of this lab and gave important feedback with respect to its applicability in university-level education.

We are planning to enhance our virtual lab with the following features:

- multi-user virtual environment and in order to allow cooperation during experimentation
- advance user interface for handling and control

ACKNOWLEDGEMENT

This work was supported by a grant of the German Federal Ministry for Education and Research under Grant No. 08 NM 101D. The authors are responsible for the content of this paper.

REFERENCES

Röhrig C., Jochheim A. (2000). Java-based Framework for Remote Access to Laboratory Experiments, In: IFAC/IEEE Symposium on Advances in Control Education, ACE 2000, Gold Coast, Australia

Bischoff A., Röhrig C. (2001). A Multiuser Environment for Remote Experimentation in Control Education, IFAC Workshop on Internet Based Control Education, IBCE'01, Madrid, Spain

Real Systems in Virtual Lab, (Web-access to our experiments)

<http://rsvl.fernuni-hagen.de>

SPS-Training (joint project with Siemens AG),

<http://prt.fernuni-hagen.de/sps/spsrail>

BMBF Project „Learnet“, <http://www.learnet.de>