

Virtual laboratory for Real-Time Control of Inverted Pendulum/Gantry Crane

H. HOYER, M. GERKE, I. MASAR, I. IVANOV, C. RÖHRIG, A. BISCHOFF

Department of Control
Prozeßsteuerung und Regelungstechnik
FernUniversität in Hagen
Universitätsstr. 27, 580 97 Hagen, Germany
Ivan.Masar@FernUni-Hagen.de

Abstract: A virtual laboratory is one of the techniques how to allow students of the distance learning gather experiences with the practical experiments. Several German universities are currently developing in the 'LearnNet' project, a network of remotely accessible virtual laboratories that will be shared by their students. We present our contribution to this project, a laboratory for inverted pendulum/gantry crane control. Through its development, several new components were used. This paper discusses the technical structure and utilization of the laboratory.

Key words: Real-time control, inverted pendulum/gantry crane, xPC Target

I. INTRODUCTION

The FernUniversität in Hagen is the only distance teaching university in Germany and an integral part of the regular public higher education system. At present, a comprehensive environment 'Virtual University' is developed in which all courses and services of the FernUniversität can be accessed and used via electronic communication and multimedia. Significant role in this concept play various virtual laboratories that enable to perform experiments with the equipment in the labs remotely through Internet. Thereby they are very useful particularly for the students of the engineering-oriented fields of study because they allow gathering experiences with practical experiments and to prove the theoretical knowledge contained in courses. Moreover, the virtual lab is one possible technique to share usually expensive equipment among several universities or education centres.

Virtual labs have been in the past years meaningful and very popular for the tutorials in control theory, since they allow performing of interesting experiments dedicated to the real-time control of various systems. At our Department of Process Control several such laboratories were established, including lab for remote control of the mobile robot with omnidirectional wheels [1],[2] and workstation for testing and programming of the industrial controllers Siemens Simatic. At the present time, we develop our latest virtual laboratory for the real-time control of an inverted pendulum/gantry crane system. This lab is a one part of our contribution to

the 'LearnNet' project [3] in which collaborate several German universities for the purpose of development of a common platform for learning and experimentation in control theory. This project will finish in 12/2003, but we expect the first students testing our lab as early as in summer term of this year.

As mentioned above, our new virtual lab allows an on-line control of the inverted pendulum/gantry crane system. This system was chosen because it represents one of the most commonly used non-linear systems in the control theory at the undergraduate level. The ability of this system to provide interesting and dramatic experiments makes it very attractive for users. Another advantage is the possibility to exploit several various control techniques for controlling of this system (PID controllers, fuzzy logic, state-space methods, etc.) and thereby to provide to students their comparative study.

II. VIRTUAL LABORATORY OVERVIEW

The virtual lab is dedicated to design and verification of several controller types and thereby to integrate the practical experimentation in control theory education. At this time, the students can try several control algorithms for various systems configuration (inverted pendulum/gantry crane/only cart position and speed control):

- State-feedback controller
- Cascade loop with PID controllers
- Fuzzy controller

For realistic presentation of the experiment, several methods were used. The students can choose from video- and audiostreams, 3D animated graphic or curve plotting, according to speed of their connection to the lab. As the interface to the laboratory, only standard Web-browser (optional with VRML Plug-In and Java Media Framework) is required.

Considering lab architecture, used components and safety requirements, the students cannot test their own control algorithms. They can choose one of the available controllers and set its parameters.

The main requirements imposed by the laboratory architecture design were:

- Modularity – decomposition of the lab on several modules with defined interface that can be alternatively used in other labs.

- Flexibility – simple components should be easily replacable with their new/improved versions and created/adapted modules should be usable in other labs, respectively. Laboratory should be upgradeable for other experiments with new algorithms in a short time with minimal effort.
- Interactivity – the students should be able to influence the flow of the experiment widely, they must not be only passive participants of the experimentation.
- Versatility - the possibility to present the experiment in several ways (multimedia, graphs, 3D-graphics), depending on the quality and speed of the user connection to the lab.

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

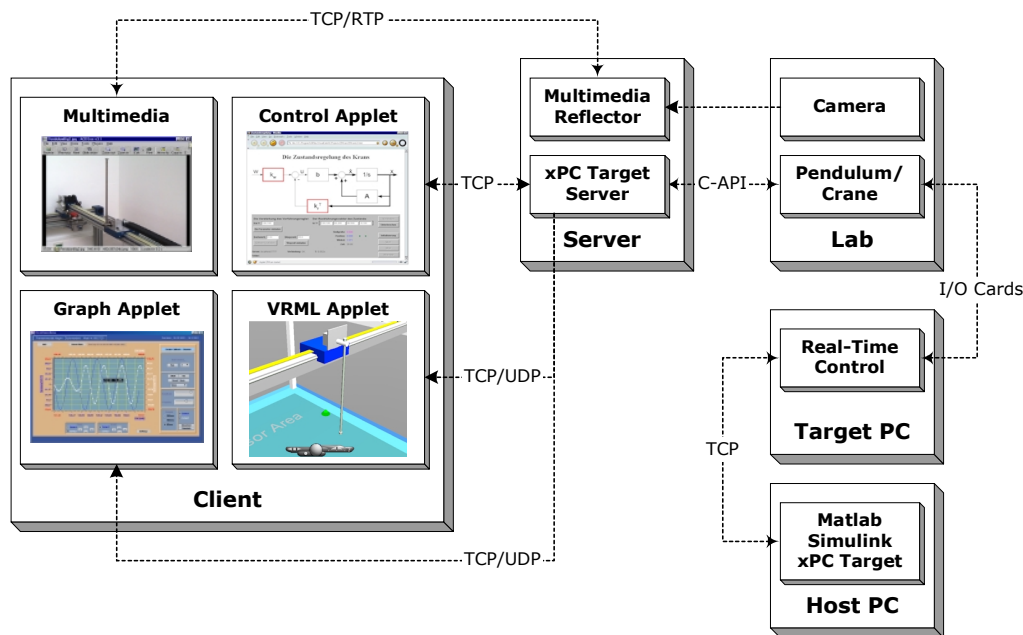


Fig. 1 Virtual laboratory components

Used/developed software components and tools used in the lab we can organize to three groups:

- Components for experiment setup (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 Target Server, Multimedia Reflector
- Client components for lab operating and experiment visualisation – Control Applet, Java

Media Framework, Graph Applet, VRML Applet

I. EXPERIMENT SETUP

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 in x-axis and pendulum/crane system movable in y-axis placed on it. The pendulum/crane subsystem consists of a cart and a rod. The DC motors drive both the gantry and the cart. The metal rod is connected to the cart by the axle and can free rotate about the z-axis. There are

three incremental optical encoders used for the sensing of the gantry position, cart position and pendulum angle. For the control, standard PC with an input/output cards is used.

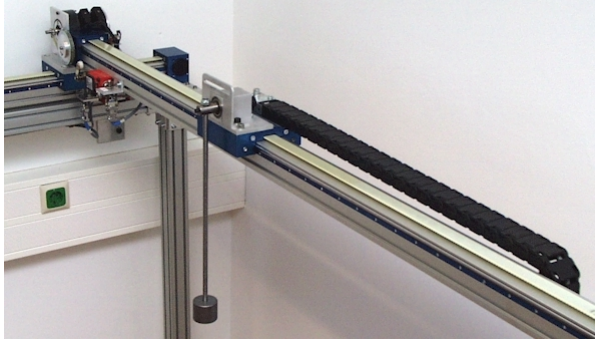


Fig. 2 Inverted pendulum/gantry crane system

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 generate automatically stand-alone real-time applications from the Simulink models that run on the so-called *Target PC*, while their development is done on the host computer (*Host PC*). Target PC runs under 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 manage the Target PC (load/start/stop application, change algorithm parameters, trace signals) fully by the Host PC (from the Matlab command line or directly from the Simulink model). The real-time application build process is fully customisable and it is possible to add own functions with algorithms written in C in the Simulink model too. The xPC Target Toolbox supports many I/O cards for data acquisition and control; however, we had to write our own code for our additional hardware (I/O cards Hesse&Knipps). The main advantages of this experimentation platform are short developing time of the control algorithms, low requirements laid on the Target PC (486/8MB PC is often suitable minimum, we used Pentium MMX/200MHz/32MB) and the stability and reliability of the Target PC since it runs with optimised operating system with only necessary functions.

II. SERVER COMPONENTS

Server components control the experiment flow (xPC Target Server) and exchange the data between the user and the lab (both xPC Target Server and Multimedia Reflector).

A. xPC Target Server

Even though the Target PC is fully controllable from the Host PC, this feasibility was used only in the development and testing phase of the lab. For the online experiments we decided to make *xPC Target Server*, which manages the Target PC and communicates with the client components on the other side. For this purpose, the released libraries with the C-API functions for the operating of the Target PC were used. These functions serve to basic Target PC control (restart, opening/closing of the communication port), setting of the real-time application and control of its flow (load/unload, start/stop, stop time and sample time setting), adjustment of the control algorithms and signal tracing.

Because the calling of the C-API functions wastes relatively much time, it was disadvantageous to use them for on-line data visualisation (with Graph Applet or VRML Applet). Therefore, we used special built-in blocks for communication through UDP communication protocol supplied with the xPC Target Toolbox in our Simulink models instead. Then, the instantaneous signal values are sent every n -th sample time through UDP protocol to the xPC Target Server, which transmits them to the client applications in desired format through UDP or TCP connection. Since UDP is connectionless, packed-oriented and unreliable network protocol, it is a high-speed type of communication and no problem arises when xPC Target Server does not receive some data (e.g. when it is busy with other type of communication). On the other hand, in this manner, received/send data samples are large enough for immediate visualisation.

For the exact experiment evaluation, other method for data storage is used. Since the data for analysis must be uniformly sampled, above-mentioned way of signal tracing is not suitable. Instead of this, we used other special blocks from the xPC Target Toolbox in Simulink models – *xPC Scopes*. These block are primary dedicated to displaying signals in numerical and/or in graphical form on the Target PC as well as on the Host PC. However, they save defined quantity of real-time data in RAM of the Target PC. These saved data are uniformly sampled with selected sample time over preset interval and they are accessible through the xPC Target C-API functions. The xPC Target loads them after experiment aborts and saves into the file, which can be downloaded by the student.

Another advantage of the xPC Target Server is, that it communicates with client applications too and therefore supervises the access to the laboratory.

Server communicates with one user at the same time and only this student can execute the experiments. The communication protocols differ according to the client application.

B. Multimedia Reflector

The *Multimedia Reflector* (MMR) is a server component, developed by the project-partner University in Bochum. It was designed for real-time processing of the video-, sound- and datastreams from laboratory and transmitting them live (synchronously) to the client. MMR can transmit streams to several clients with various transfer rates simultaneously. In addition to experimentation via Internet, it can be used by regular education process too, where the students can observe the experiment on several workstations (apart from lab). The MMR is based on a Java Media Framework (JMF) functions from Sun.

III. CLIENT COMPONENTS

Client components are dedicated to experiment flow control (Control Applet) and data visualisation (Graph, VRML and Multimedia Applets). The communication with the servers is based on the TCP (Control and Multimedia Applets) and UDP protocol (Graph and VRML applets), respectively and takes place in Web-browser. As it follows from component names, all communication interfaces with the servers are implemented by JAVA-applets. Thereby, a comfortable and easy to operate user interface was created. As the main requirement on client components, using of only standard Web-browser without additional software was considered.

A. Control Applet

The main component for virtual lab interactive access is *Control Applet*. Its version for gantry crane control by linear state-feedback controller is shown in Fig. 3. The tasks performed by the Control Applet are:

- Opening/closing connection with the xPC Target Server
- Transmitting the commands to the server in defined protocol (start/stop application, get/set parameters, etc.)
- Displaying experiment state (elapsed time, server answers, etc.) and signal values in defined intervals (cca 1s)
- Controlling of the communication status and user activity (when no activity occurs, it aborts the experiment and closes the communication)

For commands and data exchange between Control Applet and xPC Target Server, simple ASCII-Protocol was used.



Fig. 3 Web-browser with Control Applet

B. VRML Applet

This applet is used for 3D graphic representation of the experiment and for increasing of the users interactivity in the lab. This option was used because it allows realistic visualisation of the lab; and, on the other hand, it works with slower Internet connections too, because of the small amount of transmitted data. The 3D model of the pendulum/crane system, shown in Fig. 4, was created in VRML language, which is an ISO standard platform for creating and authoring interactive 3D scenes (virtual worlds). For their viewing, some of the VRML Plug-Ins for Web-browser (popular is e.g. blaxxun Contact) is necessary. In this viewer, the user can actively change the desired position of the pendulum/crane system by so-called touch sensor.

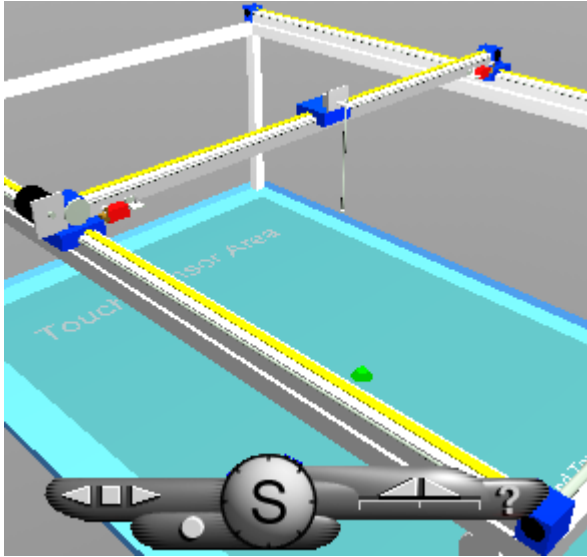


Fig. 4 VRML 3D model of the pendulum/gantry system

VRML Applet communicates with the xPC Target Server bi-directionally. The server transmits the instantaneous values of the pendulum/crane states (positions, angle, control voltage) and the applet sends back the desired position of the cart/gantry. VRML Applet animates the 3D virtual scene by using External Authoring Interface (EAI).

C. Graph Applet

Graph Applet is dedicated to immediate plotting of the experimental data. It is shown in Fig. 5.

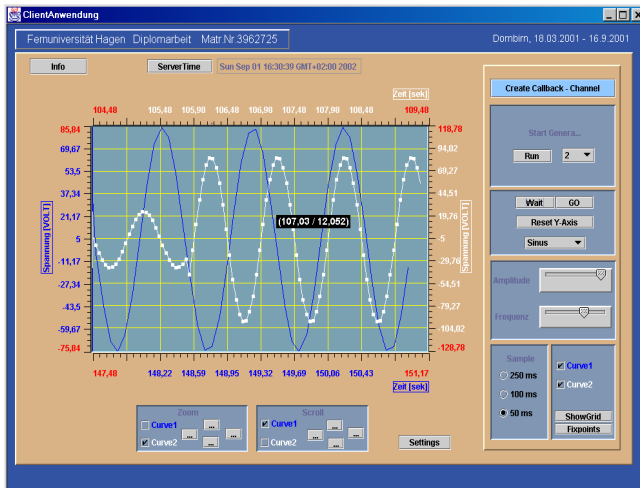


Fig. 5 Graph Applet

Graph Applet dispose of several useful functions that increase the user comfort by the signals monitoring:

- Scroll and zoom function (both independent for all curves and axes)
- Interpolation of the points between two samples
- Turning-on, turning-off the curve
- Change of the line colour and width
- Setting of the max/min values for axes

Graph applet only receives data from xPC Target Server through UDP protocol, since it is not crucial for the safe running of the experiment and the process data for exact evaluation are provided by the upload in the files.

D. Multimedia Applet

Multimedia Applet can display streaming video from the virtual lab, play sound and roughly plot the process data. This component was developed together with MMR at the University in Bochum. 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 chose the media that he wants to receive (video, sound, data) as well as bit rate for video stream. Multimedia Applet receives the data directly from MMR. One example of the video stream from our virtual lab is in Fig. 6.



Fig. 6 Video stream from the lab

IV. CONCLUSION

In this paper, concept of our new virtual lab for the inverted pendulum/gantry crane system was presented. The components used in the lab as well as on the client side of the experiment were discussed. For the real-time experimentation with our controlled system, the xPC Target Server was made. 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. On the client side, only minor demands are required. Moreover, the students can choose among several methods for experiment presentation. Students will test this new lab in the summer term, this year.

ACKNOWLEDGMENT

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.

V. REFERENCES

- [1] Röhrig C., Jochheim A.: Java-based Framework for Remote Access to Laboratory Experiments, IFAC/IEEE Symposium on Advances in Control Education, ACE 2000, Gold Coast, Australia, December 2000
- [2] Bischoff A., Röhrig C.: A Multiuser Environment for Remote Experimentation in Control Education, IFAC Workshop on Internet Based Control Education, IBCE'01, Madrid, Spanien, Dezember 2001
- [3] BMBF (German Federal Ministry for Education and Research) Project „Learnet“, <http://www.learnet.de>
- [4] Sonnweber, C.: Design and Implementation of JavaBeans-Library for Teleoperation of Laboratory Experiments, Diploma Thesis, University of Hagen, 2002