

C. Schmid

# Remote Experimentation in Control Engineering

## INTRODUCTION

One of the salient features of engineering education is the combination of theoretical knowledge with practical experience. The former, in conventional education, consists of lectures and exercises supplemented by lecture notes and textbooks, while, the latter comprises highly resource-demanding laboratory courses. The scarcity of lab facilities and staff limits the student enrolment. Active problem solving and visual feedback on the part of students can provide a valuable insight into the problems. One of the aims in control engineering education is to teach the techniques and possible pitfalls of theory-based design methods when applied in practice. The control system design process involving practical experiments and observing the dynamics of a real plant gives a valuable insight. For this reason, control engineering students have to be in a laboratory to gain hands-on experience. The development in communication techniques has opened new opportunities for students to take advantage of a remote laboratory, where an audio-visual interface together with a collaboration tool gives them the feeling of being in the laboratory.

The World Wide Web successfully demonstrates how current technology can support information sharing among widely dispersed groups. The hypertext coupled with wide area network functionality in the Web is the major reason for exploiting this technology for educational purposes to implement open and distributed hypermedia. The integration of multimedia elements has significantly enhanced the ability to train and educate electronically. Virtual education explores a new way of teaching and learning, which, instead of restricting itself only to enrolled students, addresses a larger audience. It, therefore, is capable also to be used for advanced training.

In this contribution it is shown how control engineering real laboratory experiments can be implemented and offered such that they can be performed remotely via the Web. In order to maximise the educational value of running remotely operated experiments, it is important that it as closely as possible resembles running local experiments. Some important factors in this respect are the ability to observe, manipulate and control the experiment in a variety of ways, and the ability to compensate for the remoteness by utilising advanced techniques, like video and audio communication, animation and different virtual- and augmented-reality techniques.

## WEB-BASED EXPERIMENT ENVIRONMENT

This paper is devoted to the remote laboratory approach for experimentation with real plants, which uses the communication techniques of the Web. The aim of this distance learning environment is to facilitate the access to these plants for students in control engineering. Furthermore, resource sharing among the involved universities reduces money and time for maintenance or development of new experiments.

One important aspect when designing a remote laboratory is to keep all hardware components as simple as possible to guarantee fast and effective maintenance. The same criteria apply to the utilisation of commercial available software packages. The idea is, using software and tools that are already available at most of the educational institutions and that are free on the Web. The modular structure of the remote laboratory environment proposed is shown in Fig. 1.

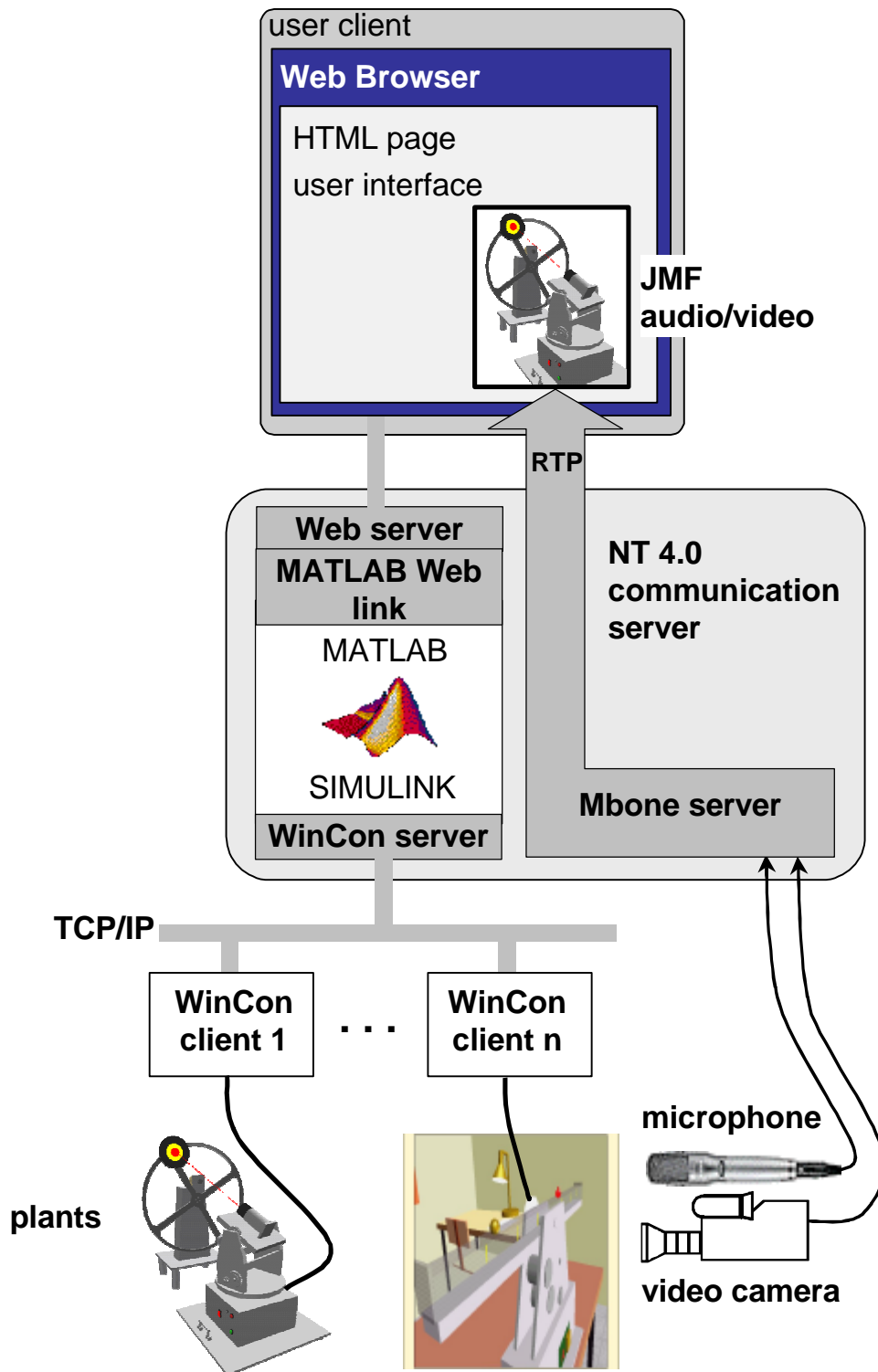


Fig. 1. Schematic diagram of the remote laboratory structure.

The *communication server* runs on a PC with MSWindows NT 4.0 as operating system and uses MATLAB/SIMULINK combined with the WinCon server to communicate with WinCon clients (real-time clients). WinCon is a product of Quanser Consulting, which uses SIMULINK and MATLAB's Real-Time Workshop for generating application code that runs in the real-time clients [1]. The communication between the Web server and MATLAB is performed through a Web link DLL using MATLAB's engine interface. MATLAB commands embedded into the Hyper Text Markup Language (HTML) page code of the user interface are sent to SIMULINK through this DLL attached to the Web server, to get or set parameters in the SIMULINK model. In order to operate the experiment, the WinCon server mirrors these actions between the SIMULINK model and the real-time client. On return MATLAB generates JavaScript code, which is transferred through the same channel back and executed in the user interface. The advantage of this configuration is that new experiments with different physical structures can be easily set up, since one can directly write new MATLAB commands embedded in the JavaScript language. On the communication server only the SIMULINK model and the corresponding M-files have to be rebuilt.

A *WinCon client* is a real-time client since it runs the real-time code corresponding to the implemented SIMULINK model. Therefore, a WinCon client contains a multi-I/O card [2], which builds the hardware connection to the plant via different types of digital and analog I/O-ports. WinCon clients run on PCs with MSWindows 95 as the operating system.

A *microphone* and a *video camera* are connected to an audio and a video capture interface, respectively, which is included in the communication server performing live audio and video transmission for trying to obtain the necessary "closeness" to the experiment. The audio/video transmission and receiving used here is based on the concept given in [3]. The Mbone server is installed on the communication server machine and it consists of the Mbone RAT audio and VIC video transmitter. They produce an audio and video stream, which use RTP UDP/IP channels [4]. On the user side the Java Media Framework (JMF) tool [5] is for displaying the video and playing the audio signals through an applet in a browser window. Fig. 2 shows this window, denominated as *Real-time video connection* including the *Real-time audio pane* for audio tuning. A resolution of 288×252 pixels, a frame rate of up to 10 frames per second and encoding in h.261 format is a good practical choice for transmission. This set-up guarantees a good animation property, since the time constants of the plant used in our experiment are in the order of 0.1 seconds. It is also acceptable for students, who are using the remote lab from home via ISDN lines.

The *user client* is a Netscape browser, which runs on the student's computer to access and operate the remote experiment. The Web page necessary to control the experiment is organised in form of a control panel, Fig. 2. Besides sliders implemented as Java applets and a graphics browser plug-in, only HTML and JavaScript have been used to organise all actions. The main operating actions (e.g. start/stop an experiment session, or reset the plant to a predefined initial state) are located in the *general control panel* frame, since these actions do not depend on the type of experiment. The specific commands to operate the plant as well as parameters to modify the characteristics of the reference signal and of the implemented controller are found in the *user console panel*. The Java sliders are used for entering continuous data and to facilitate operating the plant. Although the lack of a direct physical contact to the experiment may lead to a loss of 'practical feeling', it can be compensated by enhanced multimedia components. An *interactive plant user interface*

enables the user to operate the real plant manually by mouse actions in the virtual reality scene of the 3-D model. The actions performed with this 3-D model are immediately converted into MATLAB commands and transferred to the WinCon client. The example taken in this contribution is an optical tracking system. The 3-D plant model is written in the Virtual Reality Modelling Language (VRML). Although the user can inspect the plant by the video camera, a graphical representation of measured signals is needed to show small variations and to generate a report. The diagrams are generated remotely on the communication server by using standard MATLAB graphical routines. In order to get fast and best results, these diagrams are directly transferred and embedded on the Web page using the Extended Metafile Format (EMF). As this is not a regular Web feature, a graphics browser plug-in has been developed.

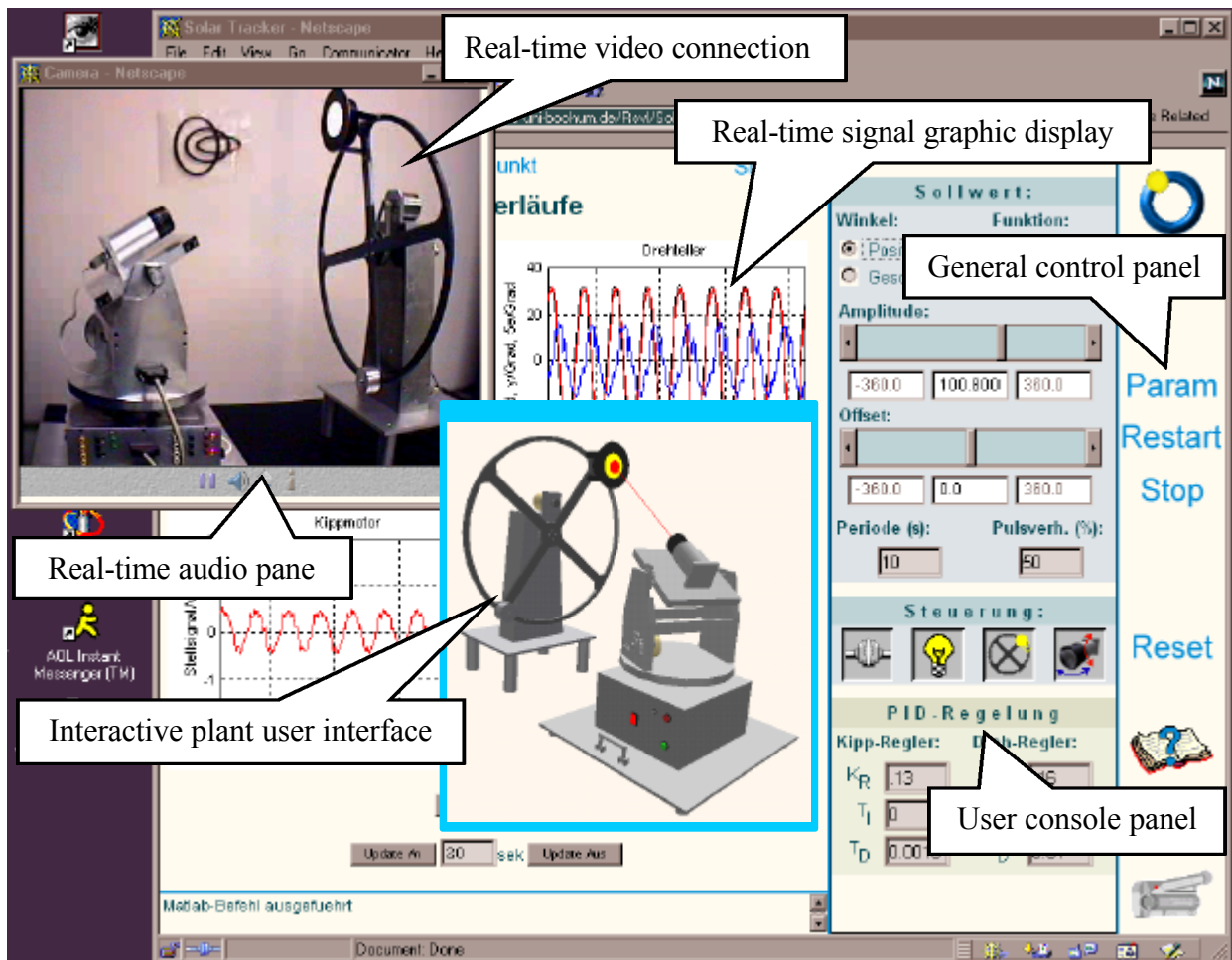


Fig. 2. Screen shot of the client main Web page including video and user interface windows.

### LABORATORY-SCALE OPTICAL TRACKER EXAMPLE

The plant is composed of two units, Fig. 3. The *reference unit* includes a light source fixed on a wheel. A DC motor moving the light spot on a circular trajectory drives this wheel. The user can select the angle (i.e. position of the light spot) or the angular velocity of the wheel (i.e. speed of the light spot). It is also possible to change the properties of the sig-

nal generator (e.g. shape, amplitude, offset, period) attained to the angle or to the angular velocity. The *tracking unit* is composed of a tracking camera mounted on a 2DOF support, which can be rotated vertically and laterally, necessary to track the light spot. Two DC motors, each of which is used to control the rotation, drive this support. The camera includes a matrix of 84 phototransistors to detect vertical and horizontal deviations of the light spot from the optical centre. The vertical deviation is fed to a controller for yaw angle control, while the horizontal deviation is fed to a second controller for pitch angle control. Both controllers are used for tracking the light spot.

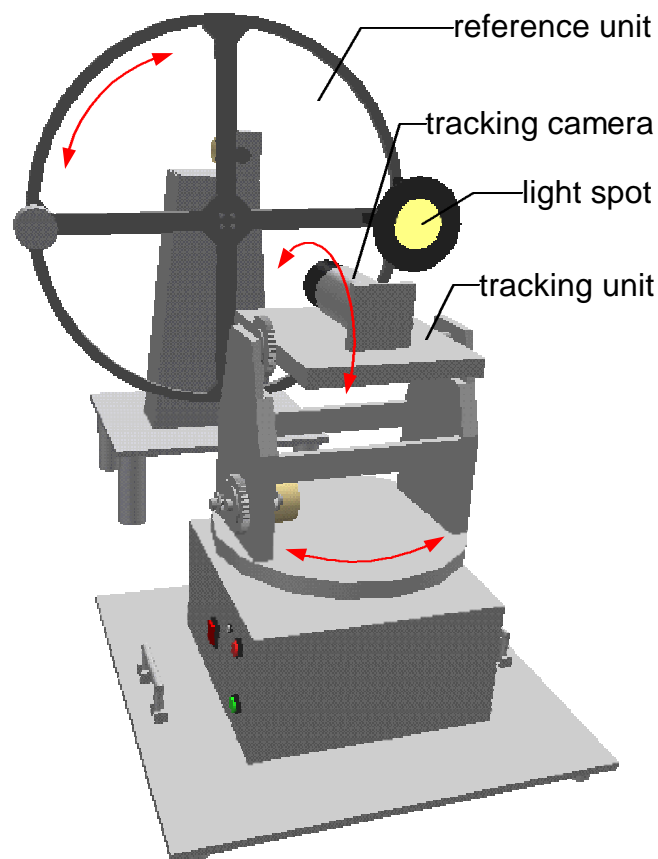


Fig. 3. Optical tracker plant (shown as VRML model)

The maximum angular velocity of the wheel is 2 rad/sec and the corresponding maximum vertical and lateral angle turns of the support for tracking the spot on the circle are  $\pm 36^\circ$ . As the tracking camera has a limitation in the view angle of  $\pm 12^\circ$ , it is a challenge for students to design a controller such that the light spot will not leave the camera's view. The controller set-up shown in the user console panel of Fig. 2 is of PID type. This experimental set-up is therefore especially suited for students enrolled in the course on basic control. The configuration can be simply changed for other type of controllers in advanced courses. The aim of the experiment above is to teach the effects of variations in the parameter of the controllers on the closed-loop behaviour. The possibilities of the configuration can be extended and range from plant modelling, system identification to servo controller design using different kind of approaches.

## CONCLUSIONS

This contribution presents a new Web-based remote laboratory environment for distance education, which addresses students and learners in control engineering. The user can perform experiments using a standard Web browser. The only requirement is to install some browser extensions, which can be downloaded from the Web server and installed simply. The experiment is controlled using JavaScript statements embedded in the Web page. They contain MATLAB commands, which are sent down to the communication server for execution in the MATLAB environment. This concept simplifies and speeds-up the implementation of new experiments due to its availability and flexibility. For an other laboratory plant only the user console panel in the Web page and the MATLAB/SIMULINK part has to be reprogrammed.

An evaluation form has been elaborated and given to students of control engineering to assess the quality of this remote laboratory experiment. Based on the feedback received, an overall positive resonance could be noticed. The quality of the control panel as well as the graphics has been approved. Controlling a real remote plant through a virtual reality 3-D model caused a surprise among the users. They have readily accepted this kind of experimentation which is available around the clock. This result has corroborated us to continue in adding other experiments and plants into this environment. Remote experiments will be standard experiments in our basic control laboratory course. From April 2000 the optical tracker plant is used in the control courses of the German open university FernUniversität Hagen. The experiment is accessible on the Web round the world and round the clock, supposed the user has been granted a login id with password [6].

## ACKNOWLEDGEMENT

The support of this project by the *Universitätsverbund MultiMedia des Landes Nordrhein-Westfalen* (Project 9-23-98 *Reale Systeme im Virtuellen Labor*) is gratefully acknowledged.

### References

- [1] Quanser Consulting Inc., "WinCon 3.0.2a Realtime Digital Signal Processing and Control under Windows 95 using Simulink and TCP/IP Technology", Canada, (1998)
- [2] Quanser Consulting Inc., "MultiQ-3tm Programming Manual", Canada, (1998)
- [3] Röhrig, C. and A. Jochheim, "The Virtual Lab for Controlling Real Experiments via Internet". Proc. IEEE International Symposium on Computer Aided Control System Design, Hawaii, 279-284, (1999)
- [4] RFC 2032 RTP Payload Format for H.261 Video Streams, Homepage: <http://www.normos.org/rfc/rfc2032.txt>, (2000)
- [5] Sun Microsystems: Java Multimedia Framework Homepage: <http://www.java.sun.com/products/java-media/jmf/index.html> (2000)
- [6] RSVL, Real Systems in a Virtual Laboratory, Homepage: <http://www.esr.ruhr-uni-bochum.de/Rsvl/>, (2000)

### Author:

Prof. Dr.-Ing. Christian Schmid  
Ruhr-Universität Bochum  
Fakultät für Elektrotechnik und Informationstechnik  
Lehrstuhl für Elektrische Steuerung und Regelung IC-3/141  
D-44780 Bochum  
Tel.: ++49 234 32 24093  
Fax: ++49 234 32 04093  
E-mail: [cs@esr.ruhr-uni-bochum.de](mailto:cs@esr.ruhr-uni-bochum.de)