

SYSID 2000
Symposium on System Identification
Santa Barbara, California, USA
June 21 -- 23, 2000

VCLAB - THE VIRTUAL CONTROL ENGINEERING LABORATORY

Christian Schmid

*Control Engineering Laboratory
Department of Electrical Engineering and Information Sciences
Ruhr-Universität Bochum, D-44780 Bochum, Germany
Phone: ++49 234 32 24093, Fax: ++49 234 32 14101
E-mail: cs@esr.ruhr-uni-bochum.de*

Abstract: This contribution presents a remote laboratory approach using virtual reality on the Web. Extensions like plug-ins and Java applets, which use MATLAB/SIMULINK for simulations, are integrated into the Web browser. Interactive animation of control engineering experiments can be performed using VRML models of laboratory plants. The Web-based techniques for realising a virtual laboratory of this kind are introduced and discussed in detail. The central focus of this contribution is put on the technical realisation of the most important elements. Some virtual laboratory examples illustrate the usefulness of this approach. *Copyright*© 2000 IFAC

Keywords: Education, Laboratory techniques, Multimedia, Software, Virtual reality

1. INTRODUCTION AND ENGINEERING EDUCATIONAL ASPECTS

The merging of hypertext and wide area network functionality in the World Wide Web (Web) is the main reason for exploiting Web technology for educational purposes to implement open and distributed hypermedia. The ability to integrate multimedia elements has significantly enhanced the ability to train and educate electronically. Whether the materials are a stand-alone tutorial or a full-fledged on-line workshop, the Web provides significant new functionality in transmitting information to the student and providing forums for exchange. The Web is an ideal tool for information which itself is rapidly changing.

Today one can observe that the scope of our technical universities is dominated by a clear-cut change. Whereas the knowledge obtained during the students' study period was hitherto sufficient for a long

period of life, today the half-life period of knowledge has been cut down such that an almost continuous updating process is inevitable until to the end of employment. This life-long learning process induces new demands and challenges on all educational institutions. This comprehends the preparation and mediation of knowledge, but also the methodical aspect of teaching and finally the organisation of the educational institution. The traditional teaching and learning methods and the organisational types give evidence of being more and more unsuitable. Traditional lectures, exercises and laboratory courses show some common shortcomings in representing networks of complex dynamic subjects. Examples, methods and tools cannot be visualised or tested by the student sufficiently in this environment. The limitation in the adaptation to the individual student and to the speed of learning at a fixed time and place is inherent. As a result of the availability of digital media, their integration and processing and due to the technical progress in communication and infor-

mation systems, one can make use of this innovative potential to establish virtual courses. By means of the independence of time and location for the access to teaching materials and teaching events the essential restrictions on conventional education are cancelled. This leads simultaneously into the transition from a lecturer-oriented to a student-oriented type of teaching. Virtual education shows a new way of teaching and learning, which is not restricted to students only, moreover, it is available for external users and, because of this, qualified for advanced training.

The World Wide Web successfully demonstrates how current technology can support sharing information among large dispersed groups. The merging of hypertext and wide area network functionality in the Web is the main reason for exploiting this technology for educational purposes to implement open and distributed hypermedia. The Web provides significant new functionality in transmitting information to the student and provides an effective mechanism for integrating tools into a single interface (Schmid 1998).

One of the salient features of engineering education is the combination of theoretical knowledge with practical experience. The first is conventionally taught during lectures and exercises and based on lecture notes and/or textbooks. The practical experience is obtained during separate laboratory courses which are resource intensive and thus it can take time and money to have innovative experiments. Engineering instruction should combine theory and practice in each lesson. The students must be able to learn e.g. modelling of systems in order to develop control systems. The design process together with simulations and observing the dynamics of a physical implementation gives the student a valuable insight. Design and visual feedback is an important aspect of engineering instruction. For this reason, students have to be in a laboratory to gain laboratory experience. Today, communication techniques become available such that students can be anywhere. This kind of telepresence may be based on telerobotic systems, virtual reality and simulation systems. A remote lab approach includes a software development tool, a video and audio interface together with a collaboration tool. This is necessary for giving the remote user the feeling of being in the laboratory. An alternative approach is to surpass the laboratory in demonstrating applications and to help the student to discover the application of theories learned in the classroom. This may be realised by the simulation of real world systems and by animation of experiments. A supplementary virtual lab during the conventional lab course and within additional distance education in the form of a course offered across the Web, will fully engage the students in the learning process through an interactive, dynamic en-

vironment involving the student, on-line materials and an instructor. The combination of these approaches in using a widespread dynamic system design software tool together with hypertext classroom materials and other hypermedia Web components leads to the concept of an integrated laboratory.

The Virtual Control Lab (VCLab) (Schmid and Müller 1997) developed for educational purpose shows these features. It is a technical realisation concept to tie together the computational and simulation machines of MATLAB/SIMULINK with Web components to a hypertext-based system, which runs on the Web browser. The main question in this context is, how to bring together Web components with simulation software and VR, with a minimum of expense in costs and programming. The preference is given to an architecture, where all computational work is performed locally. This has also the advantage that time-critical external communication problems have not to be solved and standard Web techniques are sufficient for client/server data exchange. There remain a lot of tasks to be solved for the local browser environment. One main basic sub-task is linking the MATLAB computational machine to the browser window environment, such that MATLAB statements can be embedded into HTML code for execution or that actual MATLAB graphic objects can be mapped into the page dynamically. This is necessary for any computational needs, when the pages are loaded or on interaction with the user. It is also essential to design an interface that is of general use. Bringing a VR laboratory scene onto an HTML page is a standard problem, for which browser plug-ins are available, e.g. the CosmoPlayer or blaxxun Contact VRML plug-in. In order to inspire a laboratory experiment with life, the simulation output stream has to be channelled to the objects of the VR scene in the VR plug-in. The reverse information flow is also important, when the user interacts with the experiment and the VR scene produces signals, which must be captured and converted to commands, such that the simulation model reflects the current experimental status.

This contribution deals with the techniques for solving the problems sketched above. At first the learning system aspect is discussed. Section 3 goes into a more detailed discussion of the elements of VCLab and their realisation. The virtual laboratory examples given in section 4 should illustrate the usefulness of this approach. Finally, in Section 5 the conclusions comprise the new aspects.

2. GETTING AN INSIGHT INTO LEARNING SYSTEMS

Looking at the key properties of learning systems, system flexibility and learning initiative in Fig. 1,

there are large differences between these systems. Most of the existing learning systems are completely closed systems. The content of the materials, the visual presentation, the animation or simulation and all other add-ons are generated by a higher programming language or by an authoring system. The expenses for the design of simple CBT systems is relatively low, but increases extremely with the complexity of an embedded simulation. Today one can estimate, that the author for one-hour CBT for engineering education needs 200 to 500 hours of preparation of the materials. The approach used for the VCLab is designed for decreasing these extreme costs. The general idea is to provide a common platform to this courseware system. For this reason the Web and a Web browser are chosen. This makes the system independent of a specific platform. The browser already provides general mechanisms to gather and send information through the Internet. The Web supports all infrastructures.

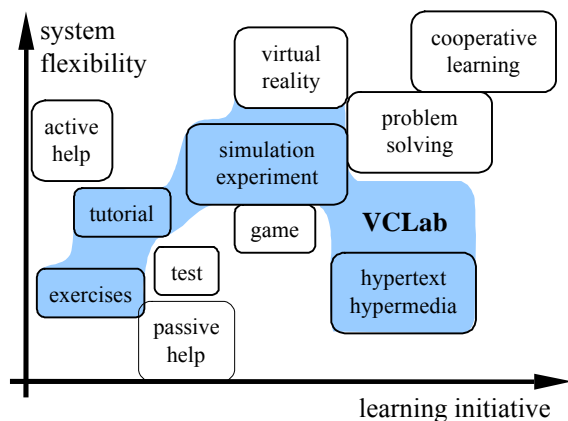


Fig. 1. Typology of learning systems

3. TECHNIQUES USED FOR THE VIRTUAL LABORATORY

The backbone language for the courseware is the Hyper Text Markup Language (HTML). On the today's standard the Web browser is capable of including Java applets into its window. This enables the author to create and use interactive and animated elements in the courseware. Powerful supplemental browser components are the dynamically loaded modules, called plug-ins. These plug-ins are used for two different reasons. First, they extend the browser's capabilities to view different file formats (animation, images, sounds, etc.). The more important part, however, is that they can be implemented in a native language, e.g. C++. This makes it possible to write applications that are capable of using all computer resources and communication methods. VCLab uses three different plug-ins, where the first two of them have been developed especially for VCLab:

1. The connection to MATLAB is performed by the MATLAB plug-in. This integrates MATLAB operation into the Web browser. The plug-in pushes computational jobs, like MATLAB statements, M-files, and pops results to/from the MATLAB engine.
2. The Graphics plug-in extends the MATLAB plug-in to view MATLAB or Metafile graphics. This plug-in pops, similarly, MATLAB graphical objects to the Web page.
3. The VRML plug-in displays 3D scenes.

The software of the plug-ins consists of Dynamic Link Libraries (DLL) and Java classes, which are responsible for embedding MATLAB into the VCLab environment and for exporting public methods. The VCLab concept makes heavy use of the Netscape LiveConnect features. LiveConnect enables communication between JavaScript and Java applets in a page and between JavaScript and plug-ins loaded on a page. Therefore, inter-object communication can be implemented in Java applets, or more simply, by using JavaScript statements directly in an HTML page. A MATLAB M-file can be hyperlinked by referencing its URL. In the HTML code fragment from Fig. 2 there is a hyperlink to „doit.m“. When loading this page, this linked M-file will be downloaded from the server and executed locally in the MATLAB workspace. This mechanism can also be applied to sets of any MATLAB files or toolboxes.

```
<html>
<body>
.....
<embed src="http://www.uni.edu/doit.m"
      name="matlab" width=50 height=10>
.....
</body>
</html>
```

Fig. 2. HTML code using the MATLAB plug-in with source tag and URL of the M-file.

With the plug-ins and these associated Java methods one has the complete control over the numerical and symbolical calculation and simulation facilities. In order to support the simulation and animation aspect and to construct a laboratory with plants in virtual reality, a supplementary set of new Web elements has been developed. This is demonstrated by the example, Fig. 3, which shows the animated simulation of a level control system with three tanks. The scene consists of a couple of different objects like lab environment with desks and chairs and the plant with three interconnected tanks, two pumps and a number of valves. The user can press buttons or open and close valves using a mouse while watching the fluid levels in the tanks. The aim is to explore the dynamical behaviour of this laboratory plant by simu-

lation studies under varying conditions. The browser is used for providing the textual explanation of the problem as well as for controlling the simulation.

The mechanical construction and the environment have been modelled by using the Virtual Reality Modelling Language (VRML). This VR scene is visualised by the CosmoPlayer VRML plug-in, which enables the user to watch the plant during the simulation from any viewpoint or angle and generates the plant's noise as well. The plug-in provides a control interface to move objects within the scene or to handle user actions while walking through the scene.

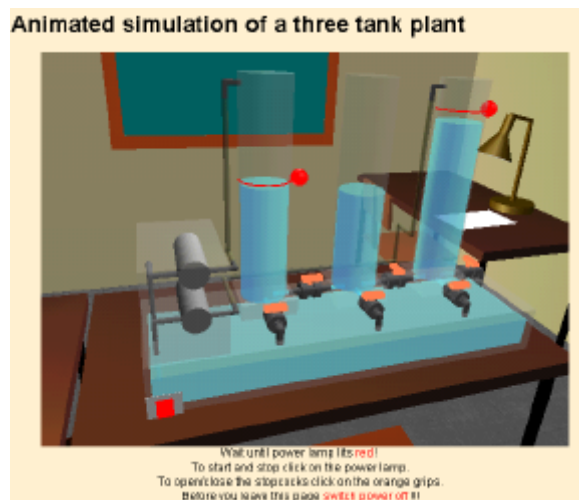


Fig. 3. Screenshot of an animated simulation example of a SIMULINK model

A SIMULINK model as shown in Fig. 4 describes the plant's dynamics. For the animation purpose an output block has been developed. This "Browser" block can be added to the SIMULINK diagram and is used for sending simulation signals to the browser and in return controls the simulation by receiving information from the browser. On the browser side this block is interfaced as shown in Fig. 5. The "Browser" block uses the socket communication techniques to communicate with the animation applet "HotAnim" via a socket S-function.

The SIMULINK/VRML map shown at the right-hand side in Fig. 5 is downloaded from the Web server together with other components. It is used to configure the HotAnim applet. This map contains the interface description between SIMULINK objects (block and variable names) and the VRML objects, actions and events (VRML node name, event name and event type) and their relationships described by properties and parameters. This includes callback actions, which apply to the user's interaction with the VRML scene. The map consists of a table of identifiers and constants, which are interpreted by HotAnim in the `init()` method of the applet. The in-

terpreter configures the internal data structures and adapts the `run()` and `observer()` method of the applet to the VRML scene and SIMULINK model.

In order to communicate with the VRML scene, the HotAnim applet uses the Java methods from the External Authoring Interface (EAI) of the VRML plug-in. The communication with MATLAB is performed by the MATLAB plug-in methods, and for data exchange and callback reaction with SIMULINK the socket-based communication line is used. In order to have a high-level interface to the VRML scene, the VRML file contains an interface written in JavaScript that is matched to the interface description of the map. The communication with this interface shows a high level of abstraction and a rich set of actions. Therefore the HotAnim applet can be configured for any SIMULINK-VRML problem.

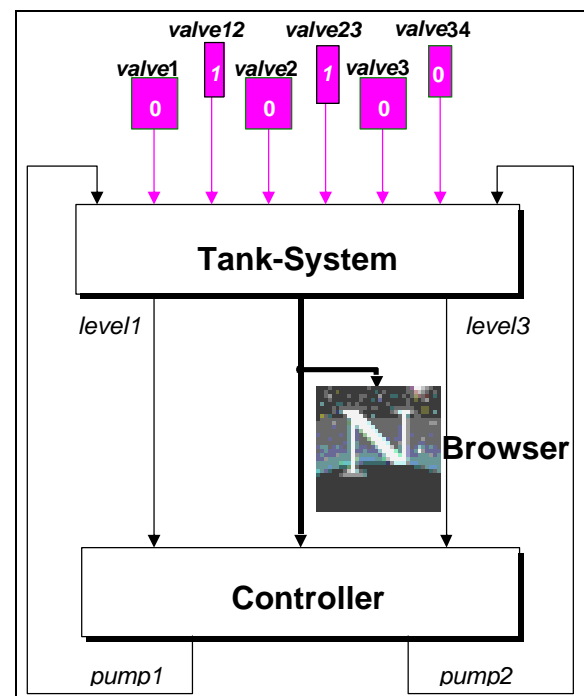


Fig. 4. SIMULINK diagram of the tank animation

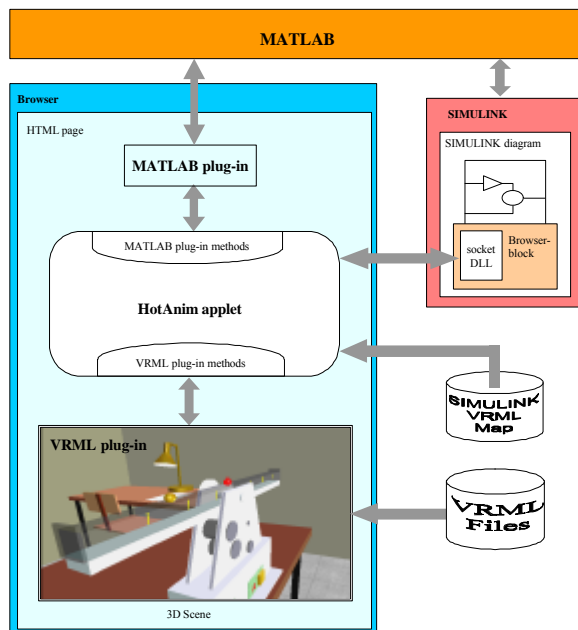


Fig. 5. Interfacing VRML scenes with SIMULINK via the HotAnim applet.

4. EXAMPLES FROM VCLAB

Some examples from the VCLab, which are based on models of real plants from the Control Engineering Laboratory at Ruhr-Universität Bochum. Fig. 6 shows the entrance of the virtual laboratory, which is



Fig. 6. A view of the entrance to the Control Engineering Laboratory in (virtual) reality.

build according to the real laboratory at Ruhr-Universität. The user can go around the floor from room to room and visit a laboratory plant at the same place as in reality. Some example plants are shown in Figs. 7 to 10. User-controlled animation technique based on simulation is applied to illustrate system dynamics of different kinds of technical processes and their control systems.

Fig. 7 shows a classical ball and beam example for teaching open- and closed-loop control. The goal is to hold the balanced ball at a specific position on the beam. The interactive elements of the scene are the power switch, the open-/closed-loop switch and a virtual ball (red ball, right ball), which acts as a slider on the beam for entering the ball set-point value in the closed-loop control task or for entering the beam angle in the open-loop case.

An intelligent machine is shown in Fig. 8, which can be mounted on the tip of a pendulum. The machine operates autonomously to raise the pendulum, to stabilise it in the up position and to perform some robotic tasks, like grabbing and transporting objects. The machine's principle of operation is based on gyration. To facilitate the development of new control methods, a development environment has been established, which consists of a MATLAB/SIMULINK simulator for the entire plant and the VRML model. A state automaton module is embedded into the system model together with all details of the mechani-

cal components and the electronic devices. New control algorithms can be designed using MATLAB and tested in this environment. For the final implementation in the embedded controller, the code-generation feature of SIMULINK can be used to generate the new controller code for the real plant.

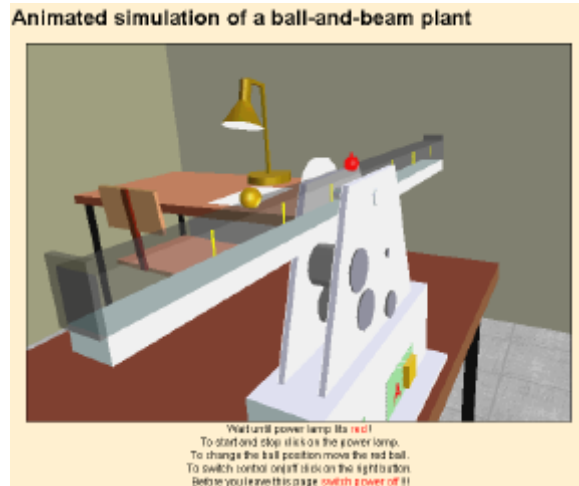


Fig. 7. Ball and beam balancing experiment.

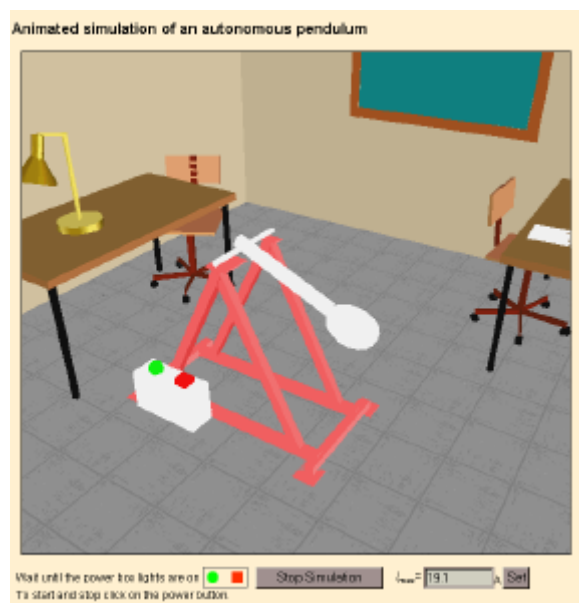


Fig. 8. Autonomous self-rising pendulum.

A similar example as the ball and beam is the VTOL emulator from Fig. 9. This unstable plant is a considerably challenging configuration for testing control operation in 3D configuration of this kind. The student can test the controller by simulation and interaction with the plant. The beam angles are given by a virtual joystick. The propellers are rotating due to the speed given by the controller. Also a variable noise generated by the motors contributes to a realistic environment.

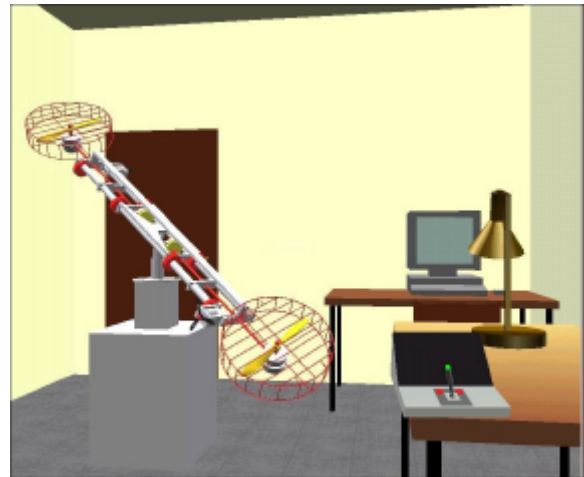


Fig. 9. VTOL control example.

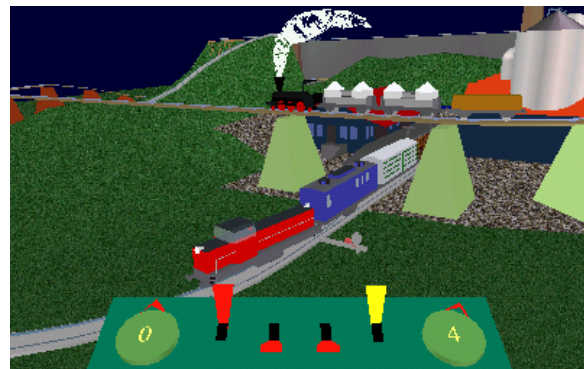


Fig. 10. Railway train control example.

The last example in Fig. 10 is a railway toy system, where the position of the speed-controlled trains can be manually controlled.

5. CONCLUSIONS

The advantage of the approach presented here is the use of existing MATLAB/SIMULINK models. E.g., only adding of one block to a SIMULINK diagram extends this tool to perform simulations in the Web environment. For engineering students it is easy to use and there is no need to become acquainted with simulation tools for using courseware based on the VCLab approach. The user can focus on the learning and understanding of problems. The usage of a local simulator shows fast simulations without the need to have high-bandwidth connections to a remote Web server. This is especially important for animation. The hypertext approach enables the authors to link different topics.

For those who want to visit VCLab:
<http://www.esr.ruhr-uni-bochum.de/VCLab/>

REFERENCES

- Schmid, C. and S.Müller (1997). A Contribution to Control Engineering Education on the Web. In *Proceedings of the 4th IFAC Symposium on Advances in Control Education ACE'97* (Istanbul, Turkey), Bogazici University, Bebek, Turkey, pp. 49-54.
- Schmid, C. (1998). Using the World Wide Web for Control Engineering Education. *Journal of Electrical Engineering*, **49**, 205-214.