

Environment for online experimentation and analysis

Yves Piguet, Denis Gillet, Nicola Tomatis and Roland Siegwart
I2S, STI, Swiss Federal Institute of Technology Lausanne (EPFL)
CH-1015 Lausanne, Switzerland
E-mail: *firstname.lastname@epfl.ch*

Abstract

Access to online experiments permitted by the current technology offers more flexibility than local presence. However, it also presents new challenges, such as which information should be transferred from the process to the remote user to describe with sufficient details the progress of the experimentation, and from the user to the process to control it and to test interesting working conditions; how to present this information; or how to optimize the learning process of students. These questions are investigated and illustrated by two online environments: remote control of a mobile robot, and remote experimentation and data analysis for automatic control.

1 Introduction

Remote access to robots or other laboratory systems is an effective way to give students more flexibility and more time to perform experiments, to avoid the multiplication of costly equipment, and to reduce the maintenance. For research, access to exactly the same set-up enables useful comparisons of different control strategies. A third benefit is to better explain what engineering is to younger students to help them decide what studies to choose.

At Swiss Federal Institute of Technology Lausanne, remote experimentation and distant learning have been investigated since 1995 at Laboratoire de microinformatique (LAMI), Institut d'automatique (IA), and Autonomous Systems Lab (ASL), which have now merged into Institut d'ingénierie des systèmes (I2S). Research performed at LAMI and ASL includes microcontroller and microprocessor boards and mobile robots; and at IA, modeling, control, and optimization.

Khepera, a miniature mobile robot developed at EPFL and made by K-Team [5], is well-suited for remote experimentation because it can be placed in a small set-up of about one square meter while still permitting interesting remote control. The first attempt to control it on the Web at EPFL was done in 1995 by writing a plug-in module for CU-SeeMe, a video-conference application used for video feedback. Its successor, KhepOnTheWeb was developed in Java and HTML (Fig. 1); it was a big success between

early 1997 and early 2000 when it was available on the Web [11]. It paved the way to a commercial development, *remotobot.net* [10].



Figure 1: The environment used for KhepOnTheWeb.

Alice, a very small autonomous robot developed at ASL, has been made available on the Web to local and distant visitors of the Swiss Museum of Transportation in Lucerne (Fig. 2) [12]. Up to five visitors can control five robots with Java applets.



Figure 2: Five Alice robots controlled by five distant and local visitors at the Swiss Museum of Transportation in Lucerne.

In automatic control, the initial purposes of the development of remote experimentation were the sharing of unique or expensive equipment among different laboratories, and demonstrations during ex cathedra courses [4], [6], [3]. Virtual instruments which had been devel-

oped with LabView (from National Instruments) for local, standard experimentation, were split into two parts: on the server, to control the system with a local feedback loop; and on the client, with graphical controls to let the user specify the experiment parameters (structure of the controller, controller coefficients, kind of reference signal) and observe the responses. Video, enhanced with synthetic images based on measurements which require less bandwidth, makes easier the supervision.

The remainder of the paper is structured as follows. Section 2 describes an online setup for the remote control of a mobile robot, with different levels of autonomy. Section 3 presents eMersion, a Web-based environment for experimentation in automatic control. Section 4 discusses the requirements of server-side software for analyzing the experimental data without any installation on the client computer. Conclusions and perspectives are given in Section 5.

2 Web-based supervision of a mobile robot

The ASL is engaged in mobile robotics research which mainly focuses on indoor localization and map building, outdoor locomotion and micro mobile robotics. In the framework of a research project on mobile robot localization, a graphical Web interface [7] for indoor robots [13] has been developed. The purpose of this interface is twofold: it enables the supervision of the autonomous robot's work and the specification of the tasks it has to perform. The graphical design has been developed focusing on the robot's sensors: the wheel encoders, a 360 degrees laser range finder and a CCD camera.

2.1 Supervision

Testing algorithms like localization and obstacle avoidance requires the evaluation of the algorithmic reactions to the machine's perception. However, fully autonomous self-contained robots close their control loop directly into the embedded system. Only the robot's behavior, not the sensor data, is accessible to the researcher. Off-line analysis can be performed by tracing the robot's position, saving all the sensor data, the extracted features, and each algorithmic result. Nevertheless, this procedure has several disadvantages:

- The correspondence between the behavior of the robot and the data which caused it is difficult to identify.
- Critical states of algorithms which may cause a failure cannot be detected before the failure.
- Crashes of one or more execution threads can cause a loss of information which could prevent a correct analysis of what went wrong.

On-line supervision (Fig. 3) is therefore proposed. Having access to the machine's perception permits to identify the correspondence between the perception and the behavior of the robot. This is done by visualizing sensory information on several levels of abstraction using state-of-the-art Web technology yielding a plug-in-free interface which can be viewed with a standard browser. It provides multi-modal information in several representations: off- and on-board vision, laser and odometry. This tool proved to be indispensable in the developing phase of navigation algorithms for localization, obstacle avoidance and path planning [13].

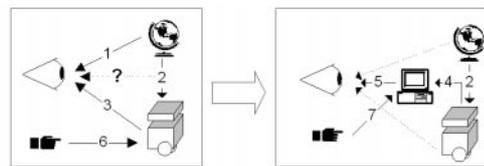


Figure 3: For mobile robotics research, the correspondence between robot behavior (3) and robot perception (2) is important. This is achieved by visualizing the robot's perception in a graphical interface (4-5). 1: Human perception. 2: Machine perception. 3: Human perception of the robot behavior. 4: On-line transfer of the machine data. 5: Visualization of the robot's data. 6: Commands via prompt. 7: Commands via graphical interface.

2.2 User interface

By performing public demonstrations of the robots, the limits of the supervision interface became apparent. The robots were controlled by a prompt, tasks were specified by text. People who were unfamiliar with the system had some difficulty to understand what was going on and how autonomous the robots were. Including task specification in a graphical feedback interface for making the results of robotics research accessible to potential end users became a new goal. This has been achieved by introducing modern guidelines for ergonomic interface design like context-sensitive popup menus or clickable goal specification (Fig. 4).

The practicability of this interface has been extensively demonstrated in the Computer2000 exhibition event, a computer trade-show held annually at Lausanne, where during four days the robot received remote commands from visitors using this interface [1]. The visitors defined 724 missions for the robot, which had to autonomously travel a total of 5.013 km in order to fulfill them. The experiment has been repeated during 10 days for *Science & Cité*, an event which has been organized in the whole Switzerland in Spring 2001 in order to make the people aware of the role of the universities.

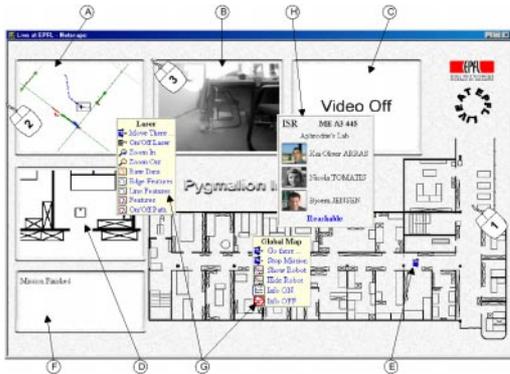


Figure 4: The Web interface. A: Multi sensor localization monitor. B: On board video. C: External web-cam. D: Local robot position (x,y,θ) . E: Robot position (x,y) . F: Message window. G: Popup menu on each window to access functions. H: Office information popup menu. Numbered mice are possible way to control the robot. Mouse 1: Set a room as new global goal. Mouse 2: Set (x,y) as new local goal. Mouse 3: assign new orientation θ .

3 Web-based experimentation in automatic control

3.1 Learning environment

The eMersion environment, a project of distant learning which involves three laboratories, currently supports pilot-courses in automatic control, fluid mechanics, and biomechanics. These courses are taught to students from several engineering majors at EPFL. Traditionally, each of these courses is partitioned into theoretical and hands-on experimentation sessions, i.e., lectures and (virtual or real) laboratory exercises. This partition, as well as the usually fixed schedule for the hands-on sessions, is a consequence of school policies and logistic matters. In fact, there are strong pedagogical reasons to avoid the decoupling of the lectures and the hands-on activities, as well as to avoid fixing the date, time, and place of experimental activities.

The aim of applying the eMersion paradigm in engineering education is to add flexibility to the students' experimental work and to better integrate complementary activities, such as lecture, exercises and laboratories. Special modules describing hands-on experimentation packages are provided online as complementary support to the lectures. Each module includes the necessary components to carry out a complete laboratory session. This mandatory course is taught at the EPFL to students from mechanical, electrical, and mechatronics engineering. The requirement in terms of the specific modules to be completed by the students varies from one major to another depending on the curriculum adopted by each discipline.

As discussed below, the eMersion environment allows students to execute the tasks of any given module remotely. Hence, the students have the possibility to carry out an experiment at a time and from a location of their choosing, and therefore benefit from a more effective cognitive experience. It is worthwhile to note that for the regular students at EPFL, i.e., those located directly on-campus, there is no formal requirement that they use the resources from locations away from the campus. In the spirit of flexible learning, the students are in fact also allowed to carry out the experimental work directly on the campus premises; however, access to the facilities is restricted to a number of specific time slots. Consequently, the experimentation modules have been designed to ensure that they are equally helpful to local and to remote students. The modules introduced into the different pilot-courses have been carefully designed and the pedagogical scenarios of the courses have been revised and adapted in order to maximize the benefits realized via the integration of the online modules.

In automatic control, the challenge has always been for students to link the highly mathematical analysis and design methods introduced during the lectures with the actual implementation of feedback loops. To facilitate this association process, an inverted pendulum and an electrical drive are introduced as physical experimentation setups. The former is used by the teacher for demonstration purposes, and the latter by the students for hands-on experiences. Instead of devising a single online module for each of the former 'classical' experimental sessions, it was decided to offer a larger number of modules which in turn are shorter in required execution time, and also more focused in terms of the objectives. Care is taken to ensure that the modules allow the students to learn one topic at a time, rather than being overwhelmed with a large number of concepts during a lengthy session. Another key change in implementing the eMersion paradigm has been the complete decoupling of both the teaching-assistant support and the evaluation task from the experimentation activities. Tutors now replace the teaching assistants that used to grade the students' work under the previous scheme. The students either can contact the tutors directly during office hours, or asynchronously by e-mail. Currently, no grades are given for the timely completion of the modules during the term. The only constraint for the students is to successfully fill a prelab form that poses technical questions that must be answered to obtain the permission to access a given module. This requirement has been introduced to ensure the students have the necessary prior knowledge to benefit from the experience, and to motivate them to do preparatory work on their own. A grade is given only at the end of the term, when every student is assigned a randomly selected module that they are required to execute within a pre-specified time. The existence of a test module at the end of the semester is, in our experience, a strong motivating factor to induce

the students to learn as much as possible during the term, and to develop a thorough understanding of all the freely accessible modules.

A Cockpit metaphor has been introduced as a graphical user interface (GUI) to integrate all the components necessary to complete successful interactive experiments and to sustain collaborative activities between the students. As an example, Fig. 5 shows the cockpit that is available to complete an experimentation module in automatic control. In this case the EM is dedicated to the control a real electrical drive that is visualized within the central video frame of the cockpit.

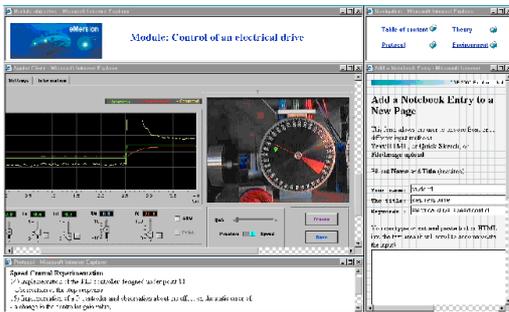


Figure 5: The eMersion cockpit for the control of an electrical drive.

In the eMersion environment, the interaction relies mainly on changes made by the students in the configuration algorithms or parameters (actions on the virtual broomstick) that induce modifications in the real or virtual experiment. The modifications are graphically display as soon as possible in the cockpit (perception through the porthole or using panel instruments).

4 Data Analysis

In order to help them to prepare the experiments and to analyze the results they obtain and see how they agree with the theory, students have access to software applications. Trade-offs between openness, where the student is free to investigate many approaches, and a guiding user interface, where he or she is conducted to the solution along a well-marked reasoning path, must be found.

4.1 Software requirements

In automatic control like in many other engineering areas, Matlab (from The Mathworks) is among the most widely used mathematical software. Its language is well-suited for linear algebra and numerical computation. Libraries of functions (“toolboxes”) extend the base functionality in domains such as classical control and system identification. Matlab has been used for more than 10 years at Institut d’automatique of EPFL, by researchers as well as by students for their lab assignments and projects.

Students should continue to have access to this kind of software. However, a Web environment adds several requirements:

- Access to server processing should be made available via two kinds of user interfaces: with forms to lead the student to the solution while allowing to change parameters or experimental data, or by entering arbitrary code fragments to perform any kind of computation and creating any graphics.
- Students should have access to the software from any computer without the necessity to install additional software; a Web browser should be sufficient.
- Security should be enforced in both ways: to protect the server against attempts (voluntary or not) by the remote user to break into it, to crash it, or to consume excessive resources; and to protect the client against a server he or she does not necessarily trust.
- Integration with other Web resources (with the Cockpit in the case of eMersion) should be as good as possible, to benefit from online documentation, merging between exercises and the software tools to solve them, dynamic figures, etc.
- The work overhead for the teachers and webmasters should be as low as possible; code reuse should be maximized.

4.2 Software choice

Matlab cannot fulfill all these requirements. While there exists a server for Web applications, named Matlab Web Server, it is virtually impossible to let the remote users type commands directly and create graphics in an easy way. The function `eval`, which evaluates arbitrary code, does not disable potentially dangerous functions such as those which give access to the file system or to the shell. WebMathematica (from Wolfram Research), based on the symbolic math software Mathematica, while it makes easier the integration of graphics, suffers from the same problem; in addition, giving access to the command-line interface of Mathematica to remote clients is explicitly prohibited by the license.

For these reasons, eMersion relies on SysQuake Remote, based on SysQuake (from Calerga) [2]. SysQuake has a mathematical language compatible with Matlab. It offers interactive graphics, i.e. the capability to update the graphics in real-time when the user manipulates them with the mouse to change the initial conditions of a simulation or the parameters of a controller design [8]. At EPFL, it has been used for four years in automatic control, signal and image processing, and mobile robotics. SysQuake Remote is a module for the Apache Web server which interprets Matlab-compatible code embedded in

HTML pages. Graphical commands create images stored temporarily on the server which are included in the document, in a transparent way for the author as well as for the remote user. On the client side, the browser receives plain HTML documents with static PNG images.

User interaction does not reach the level of SysQuake, because any request from the user must be sent to the server before a completely new document is sent back. This forbids the mouse dragging of graphical elements. However, the remote user can:

- fill a form or modify its contents;
- upload the contents of a file (such as experimental data);
- click into an image (the mouse coordinates are translated to the coordinates of the graphics, taking the scale into account).

SysQuake Remote provides functions to facilitate these operations and use the information as input for its computation.

4.3 Remote code execution

The remote execution of arbitrary commands entered by the user presents some important security problems. The user must be prevented from

1. having access to most resources on the server (files, printers, network, etc.)
2. submitting jobs which take too much time and slow down the whole system
3. posting or requesting too much data

Point 1 is addressed by evaluating all the untrusted code in a *sandbox*, i.e. an execution set-up where only a restricted set of commands is allowed. Commands which may cause harm are disabled at the level of the byte-code interpreter of SysQuake Remote. These commands include file-related functions and Unix shell commands. Limits on computation time and on the amount of input and output data are also enforced in order to address points 2 and 3.

Fig. 6 shows the main HTML form used to enter commands. In addition to a text field, there is the option to upload a MAT-file (the native data file format of Matlab, also used as an exchange format for experiment data.) Commands may include expressions, assignments to variables, loops, and graphic commands. When the user click the “Execute” button, the contents of the form and of the MAT-file (if one is specified) are sent to the server where SysQuake Remote evaluates the code and produces a new

page with the previous form contents (so that the user can edit the code or correct the errors) as well as any result (text or graphics) produced by the commands.

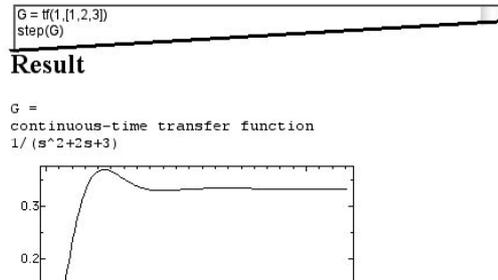


Figure 6: Text field where commands are entered (top) and result produced by SysQuake Remote (bottom). In this figure, buttons to control execution and the upload of MAT-files have been removed to take less space.

4.4 Frequency-domain identification

One of the topics students should master is the relationship between a physical system, the transfer function used to model it, and its frequency response (the system, an electrical drive, can be adequately represented as single-input single-output linear time-invariant). The experiment they should perform is the following:

- (remotely) apply to the system an open-loop or closed-loop control signal whose spectrum covers the bandwidth of interest;
- compute the frequency response of the system using FFT;
- display the frequency response graphically;
- determine the structure of a model and identify its parameters;
- check that the frequency response of the model matches the experimental one.

The first step produces a MAT-file which contains the samples of the reference, control, and output signals with the corresponding time values. At this stage, students could use the interface for remote code execution described above. In order to help them get a result quicker, the user interface shown in Fig. 7 has been developed. Students can directly choose the structure of the model (model order, delay) and enter values for the parameters (the most important ones are the static gain and the first time constant).

Frequency-Domain Identification

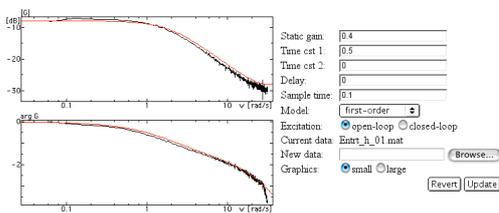


Figure 7: User interface for the computation and the display of the frequency response of a system based on an experiment and for the comparison with a linear model.

5 Conclusions

In this paper, online experimentation environments were presented which permit students to supervise mobile robots, conduct experiments to collect data and analyze these data with a friendly graphical user interface or scripts they develop themselves. Two main objectives were reached: flexibility (as few constraints as possible are imposed on how students perform their experiment), and ease of client set-up (i.e. no set-up at all; a Web browser is all that is required).

While they are two distinct environments, the remote control of a mobile robot and the remote experimentation for automatic control are or will be involved in the project Hands-On Mechatronics, whose aim is to develop a problem-based learning environment for mechatronics [9]. Through a mobile robot competition, teams of students develop various skills in mechatronics, such as the choice of sensors, the processing of the data they provide, the mechanical and electrical design of parts of the robot (the base itself is provided), the low-level motion control and the higher-level strategic control. The competition is attended on a voluntary basis; students mention as the top motivation factors the competition, the robot design, the application of theory to real problems and the project-level development. Some aspects of the project may be used to obtain credits.

6 Acknowledgment

The eMersion and Hands-On Mechatronics projects are funded by The Board of the Swiss Federal Institutes of Technology in the framework of its New Learning Technologies (NLT) program.

References

[1] K. O. Arras, N. Tomatis, B. Jensen, and R. Siegwart. Multisensor on-the-fly localization: Precision and

reliability for applications. *Robotics and Autonomous Systems*, 34:131–143, 2001.

[2] Calerga Sàrl. <http://www.calerga.com>.

[3] D. Gillet, C. Salzmann, and H. A. Latchmann. Advances in remote experimentation. In *Proc. of 1995 American Control Conf.*, Chicago IL, June 2000.

[4] D. Gillet, C. Salzmann, R. Longchamp, and D. Bonvin. Telepresence: An opportunity to develop practical experimentation in automatic control education. In *Proc. of 1997 European Control Conf.*, Brussels, Belgium, July 1997.

[5] K-Team SA. <http://www.k-team.com>.

[6] H. A. Latchmann, C. Salzmann, S. Thottapilly, and H. Bouzekri. Hybrid asynchronous and synchronous learning networks in distance education. In *International Conference on Engineering Education*, Rio de Janeiro, Brazil, 1998.

[7] B. Moreau, N. Tomatis, and K. Arras. Multimodal web interface for tasks supervision and specification. In *SPIE Telemanipulator and Telepresence Technologies VII*, Boston MA, 2000.

[8] Y. Pigué, U. Holmberg, and R. Longchamp. Instantaneous performance display for graphical control design methods. In *IFAC*, Beijing, China, July 1999.

[9] Y. Pigué, F. Mondada, and R. Siegwart. New environment for learning by doing in mechatronics education. In *Workshop on Robotics Education and Training*, Weingarten, Germany, July 2001.

[10] remotebot.net. <http://remotebot.net>.

[11] P. Saucy and F. Mondada. KhepOnTheWeb: One year of access to a mobile robot on the internet. In *Workshop WS2, Robots on The Web, IROS'98*, Victoria, Canada, October 1998.

[12] R. Siegwart, P. Balmer, C. Portal, C. Wannaz, R. Blank, and G. Caprari. Robonweb: A setup with mobile mini-robots on the web. In K. Goldberg and R. Siegwart, editors, *Beyond Webcams, an Introduction to Online Robots*, pages 117–136. MIT Press, Cambridge MA, 2002.

[13] N. Tomatis, R. Brega, K. Arras, B. Jensen, B. Moreau, J. Persson, and R. Siegwart. A complex mechatronic system: from design to application. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Como, Italy, 2001.