WEB-BASED LEARNING RESOURCES FOR VOCATIONAL TRAINING FOR AUTOMATION TECHNICIANS

H. Vargas, J. Sánchez, S. Dormido, G. Farias, N. Duro, R. Dormido-Canto, S. Dormido-Canto, F. Esquembre*

Dpto. de Informática y Automática, UNED, C/ Juan del Rosal nº 16, 28040 Madrid. Spain.
E-mail: hvargas@bec.uned.es
**Dpto. De Matemáticas, Univ. de Murcia, Spain*

Abstract: This paper presents the state of development of two innovative web-based control laboratories and focuses in particular on the needs of the automation technician community. The two processes selected to be simulated and controlled via the Internet were a self-regulating liquid level process and a didactical heat-flow setup. Both are good representations of many of the processes encountered in process control. The client-side of both virtual/remote labs was developed using the programming support provided by *Ejs*, an open-source tool for generating powerful Java applications and applets where knowledge of advanced programming is not necessary. *Copyright © 2006 IFAC.*

Keywords: laboratory education, distance learning, web-based experimentation, virtual laboratory

1. INTRODUCTION

Automation technicians play an important role in the day-to-day operation and maintenance of automation equipment and control systems in all process and manufacturing industries. Automation technicians do acquire basic understanding and knowledge of automation and control systems in the initial vocational training in automation at an educational institution. In spite of this, there is a documented need for specialized and ongoing training and practice in addition to basic education. This is also due to the organization of many companies:

- Automation technicians often have less opportunity of attending training courses etc. to update their skills compared with university graduates.
- Even in large companies there is often only a very small group of people with an automation and control background. This means that automation technicians do not have much opportunity of discussing problems related to their profession with in-house personnel.

Industry constantly puts pressure on educational institutions to produce automation technicians with sufficient basic knowledge and experience so that in-company formal and/or non-formal learning is kept to a minimum. Consequently, the need is to provide learning materials and contents that are relevant to industry and continually updated and modified to match industrial equipment.

To provide some solutions to these problems, five European universities are running a pilot project entitled "AutoTECH: Automation Technicians Vocational Training Repository" which is part of the European "Leonardo da Vinci" programme (AutoTech, 2006). The specific aim of the project is to develop and disseminate a set of new and innovative software packages for vocational training in automation and control. The training packages will improve quality and increase motivation in vocational training. Each of these packages will consist of different types of web-based learning resources: theory, exercises, interactive simulators, remote operable labs. As a result of this project, automation technicians will acquire knowledge and practical experience in the design, tuning, and troubleshooting of industrial control systems.

The aim of this paper is to highlight the quality and relevance of the learning materials developed in this project. To achieve this objective, two of the virtual and remote laboratories created by one of the members of the AutoTech consortium, the UNED team, will be described in this work.

The paper is organized as follows. Section 2 describes Easy Java Simulations (*Ejs*) principles, the software package used to create interactive simulations and remote labs. Section 3 presents the structure, design, and features of the heat-flow remote/virtual lab. Section 4 discusses the design and implementation of the virtual lab to study the control of a first-order lag process. Section 5 gives some suggestions of control experiences using both web-based labs. Finally, Section 6 contains concluding remarks and considerations about further works.

2. EASY JAVA SIMULATIONS PRINCIPLES

Easy Java Simulations is a freeware, open-source tool developed in Java, which is specifically designed to create interactive dynamic simulations (Esquembre, 2004). *Ejs* was originally designed to be used by students for interactive learning under the supervision of educators with a low programming level. However, the user needs to know the analytical model of the process and the design of the graphical view in detail.

Ejs architecture stems from the model-view-control paradigm, whose philosophy is that interactive simulations must have three parts: the model, the view, and the control. Accordingly, the steps for building an application in *Ejs* are the following: (1) To define the model it is necessary to specify the variables that describe the system and the mathematical equations interrelating them; (2) define the view in order to represent the states of the process; and (3) define the control in order to describe the actions that the modeler can execute above the simulation (Figure 1).

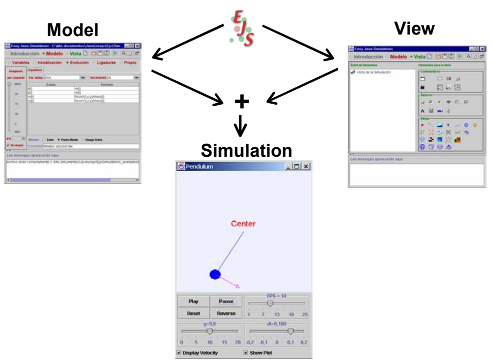


Fig. 1 Steps for creating a simulation in *Ejs*

These three parts are interconnected as the model affects the view and the control actions affect the model. Finally, the view affects the model and the control, because the graphical interface can contain information on them.

Ejs implements its particular procedure to create specific interactive simulations since it removes the control part. Consequently, *Ejs* simulations are created by specifying a model to be run by the *Ejs*' simulation engine and by building a view to visualize the model state that readily responds to user interactions. Thus, to define the model in *Ejs*, it is necessary to identify the variables that describe the process, initialize them, and also describe the mathematical equations that generate the model. The recent releases of *Ejs* support using Matlab/Simulink, Sysquake and Scilab to describe and simulate the model (Dormido *et al.*, 2005).

The view is now the user-to-model interface of interactive *Ejs*. It provides a visual representation of the model's relevant properties and dynamic behavior, and also facilitates the user's interactive actions. *Ejs* includes a set of ready-to-use visual elements. With them, the modeler can create a sophisticated view in a simple, drag and drop way (Figure 2). The properties of the view elements can be linked to the model variables, producing a bi-directional flow of information between the view and the model. Any change in a model variable is automatically displayed by the view. Reciprocally, any user interaction with the view automatically modifies the value of the corresponding model variable.

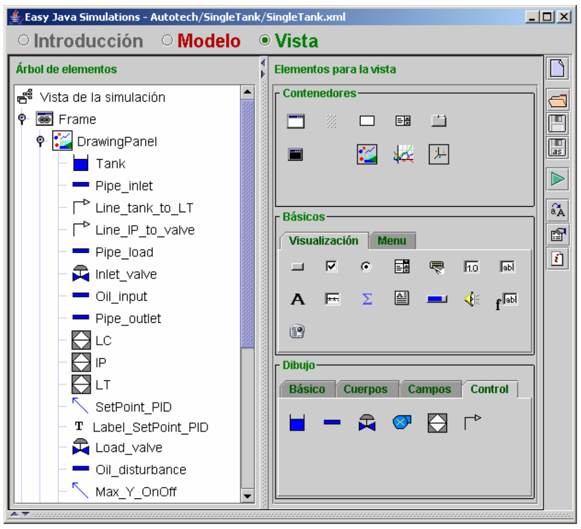


Fig. 2 Graphical user interface of *Ejs* for creating the view of a simulation. The tree on the left-hand side corresponds to the view in Figure 6.

Once the modeler has defined the model and the view of the interactive simulation, *Ejs* generates the Java source code of the simulation program, compiles the program, packs the resulting object files into a compressed file, and generates HTML pages containing the narrative and the simulation as an applet. Since *Ejs* is a Java code generator, the developer has the possibility of writing its own Java code and external libraries (by .jar archives). This feature means that *Ejs* can be used to develop remote labs because the view can be connected via the Internet by introducing the Java code to manage the connection with the remote server.

In recent years, *Ejs* has been used to develop virtual/remote labs for higher education and research and has been shown to be suitable for this (Buccieri *et al.*, 2005; Dormido *et al.*, 2005; Duro *et al.*, 2005, Pastor *et al.*, 2005; Sánchez *et al.*, 2005). For these reasons, this tool was selected by the UNED team as the prime software tool to develop the AutoTech web-based labs.

3. THE HEAT-FLOW WEB-BASED LAB

The didactical setup selected to build the virtual and remote lab was the Heat-Flow system by Quanser Consulting (Figure 3).

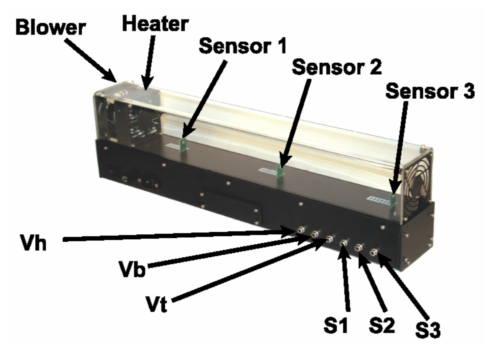


Fig. 3. The heat-flow apparatus by Quanser

The plant consists of a duct with the following components: a heating element and a blower located at both ends of the structure and three temperature sensors S1, S2, and S3 along the duct. The power delivered to the heater is controlled using an analog signal. The fan speed can also be controlled using an analog signal. Fast settling platinum temperature transducers are used to measure the temperature. Fan speed is measured using a tachometer and can be used to design speed controllers.

To model the plant, the following transfer function applies:

$$G(s) = \frac{C_n(s)}{V_q(s)} = \frac{K_p(1+sT_3)}{(1+sT_1)(1+sT_2)} e^{-Ls}$$

$C_n(s)$: Temperature at sensor n.

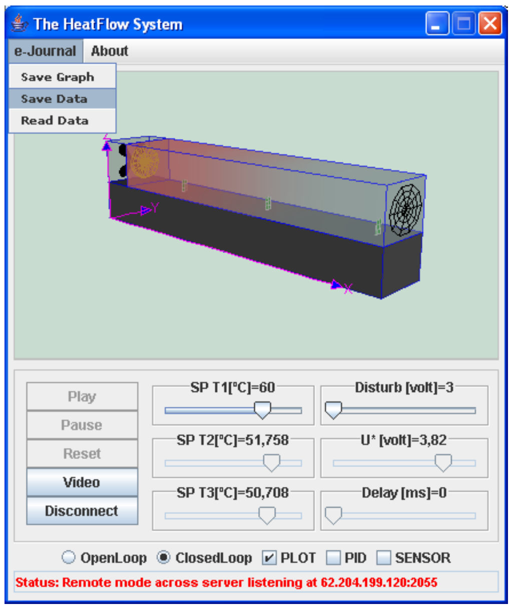
$V_q(s)$: Voltage applied to the heating element.

where steady-state gain, lags, and delay depend on which of the three sensors is used to close the temperature control loop.

3.1 THE CLIENT-SIDE

Figure 4 shows the main window of the virtual and remote. At the top of the window there is a 3D representation of the heat-flow apparatus whose color varies according to the state of the process (a model when running as a virtual lab, a real didactical setup when the remote option is used).

At the bottom of the main window, there is a control panel that allows the user to choose the type of experience (virtual or remote), switch the control mode (open or closed loop), and a set of sliders and buttons to change the operating condition (for instance, changing the set point or introducing some disturbance by varying the heater voltage).



SP T1[°C]=60
Disturb [volt]=3
SP T2[°C]=51,758
U* [volt]=3,82
SP T3[°C]=50,708
Delay [ms]=0

○ OpenLoop ● ClosedLoop ☑ PLOT ☐ PID ☐ SENSOR

Status: Remote mode across server listening at 62.204.199.120:2055

Fig. 4 Graphical user interface (*Ejs* view) developed with the *Ejs* graphical library

The four buttons located on the left side of the control panel (*Play, Pause, Reset*, and *Remote*) allow users to specify how the system must work. With the *Remote* button the user has access to the real heat-flow system running in the university lab (this feature is described in Section 3.2). If the user does not push the *Remote* button, the application works in simulation mode. Both simulation and remote results are similar because the simulation represents the true behavior of the real system. Next to the buttons are six sliders to set the values of the set-point of each temperature sensor (*SP1*, *SP2*, and *SP3*), the voltage supply to the blower (*U**), the voltage applied to the heater, and to slow down the simulation (*Delay*). With two radio buttons, *OpenLoop* and *ClosedLoop*, it is possible to select the operation mode regardless of whether the user is operating with the simulation or the real plant. If the closed-loop mode is on, the slider *U** is enabled to change the voltage. Otherwise, the slider *U** will be disabled and the value will be calculated by the PID controller.

Additionally, there are three check buttons to open the auxiliary windows. By selecting *PLOT*, a window with two scopes is displayed presenting the changes in the main variables (temperatures at the sensors, control action *U**); by selecting *PID*, another window is shown with the parameters of the PID controller associated with each sensor. The *SENSOR* button lets users choose a sensor to close the temperature control loop. Finally, there is a pop-up menu with commands to save the changes in the system variables in a GIF image (*Save graph*) or in a plain text file (*Save data*), and to open an information panel to visualize messages with the state of the connection between the client-side (the *Ejs* view) and the server side (the didactical setup).

3.2 THE SERVER-SIDE

The remote lab architecture is based on a single client/server structure, i.e., the same computer acts as a web server and controller (Figure 5). The interface of the client-side is the *Ejs* interactive view that is able to act either as a virtual lab by using the mathematical model, or as a remote lab making a TCP connection with the controller running at the server-side. This switch is done when a user pushes the *Remote* button in the view. From this moment, every user action sends a data array to the server, and a vector with the current state of the plant is obtained as a response. The control loop thus governs the heat-flow system according to the data packets sent from the view.

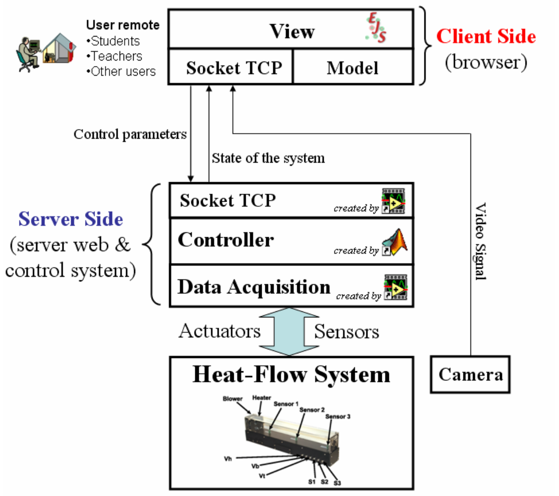


Fig. 5 Architecture of the server-side

The control parameters sent to the controller are: control mode (manual/automatic), PID parameters (Kp1, Ti1, Td1, Kp2, Ti2, Td2, Kp2. Ti3, Td3), blower manual voltage (Ub), and temperature set-points (SP1, SP2, SP3). The information returned to the view has five values: sampling time (t), temperature measured by sensors (S1, S2, S3), and voltage supplied to the heating element (Uh). The total length of the data packet is 20 bytes.

The control loop at the server-side was developed with LabVIEW and Matlab/Simulink. The data exchange between the computer and the heat-flow apparatus is done by a National Instruments DAQ, and the control loop and the socket connections are programmed by LabView. However, the PID controllers are Matlab/Simulink code. The LabVIEW and Matlab exchange the controller's information using a LabVIEW block to call and evaluate Matlab scripts from LabView programs. This mixed structure is more flexible, and new control strategies can thus be included in the server-side with little programming effort (for instance, predictive or fuzzy controllers).

4. THE FIRST-ORDER LAG VIRTUAL LAB

Another category of learning resources developed in the framework of the AutoTech project is the dynamic simulator. The aim of these simulators is to provide students with powerful but simple tools to compare and contrast many of the ideas and concepts gathered in the theoretical modules. Since the UNED group is responsible for developing the training package to study PID controllers in a broad sense, several interactive simulators of first- and second-order lag processes controlled by PID controllers were created to help students study the theory and complete exercises and quizzes. The simulator presented in this section is a learning resource designed to support the study of first-order lag processes controlled by on/off and PI controllers.

As well as the heat-flow view, the complete simulator was developed by *Ejs*. Thus no additional tools are needed to run the application except a Java-enabled web browser, since the simulator is an applet to be downloaded from the AutoTech web site.

4.1 MODEL OF THE PROCESS

The process modeled by the *Ejs* built-in equation editor is a self-regulating process: an oil tank (diameter = 12.5 m, height = 2.8 m) with an inlet valve to control the input flow and a smooth tubing acting as a restriction. The resulting transfer function of the process is:

$$\frac{H(s)}{Q_{in}(s)} = \frac{1}{1+117s}$$

H(s) being the level of the tank as a percent of full-scale range of the output (2.8 meters), Q_{in}(s) the input flow rate as a percent of the full-scale range of the input (0.4 litres/seg), and $\tau = 117\,\text{min}$.

Three basic controller modes have been included to regulate the level of the tank: two-position, proportional, and proportional-integral. The on/off controller includes a neutral zone that can be configured by the user. This feature lets students observe the impact of the hysteresis in the oscillation of the controlled variable (the level) regarding the limits of the differential gap.

As well as the tank, a continuous PI controller was modeled by the built-in *Ejs* equation editor. To obtain this, a state space model of a PI controller was used:

$$\dot{x}_c = y_{sp} - y$$

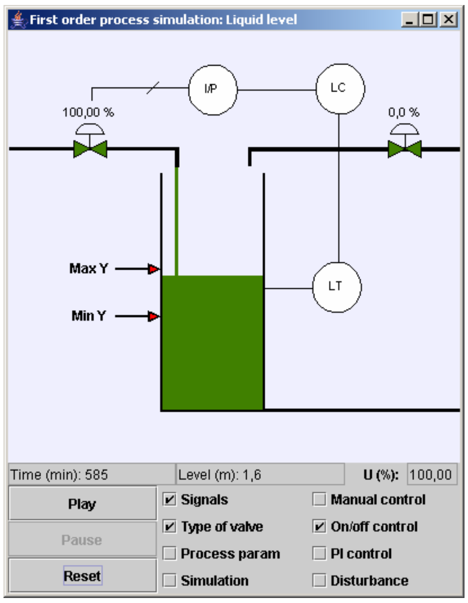
$$u = \frac{K}{T_i} x_c + K y_{sp} - K y$$

Since these learning resources are for automation technicians, the PI and PID controllers of the web-based labs do not include advanced concepts such as anti-wind-up mechanisms, set-point weighting, filtered derivative, and bumpless transfer between manual/automatic mode or between controller parameters.

To include extra features in the simulation, i.e., load disturbances, a second inlet was included in the model. It is also possible to change the type of valve used to regulate the input flow: linear, equal percentage, and fast-opening. Just the static characteristic of the valve was considered, ignoring the dynamics (it would raise the order of the process and only first- and second-order processes are included in the syllabus). However, the introduction of nonlinear valves will highlight to students the need for making use of other simple but robust control techniques (for example, gain scheduling) to cope with special operating conditions (for instance, non-linearities).

4.2 THE VIEW

Figure 6 presents the main window of the first-order lag virtual lab. It has two parts: the process diagram and the control panel. The process diagram is dynamic and interactive since it allows users to modify references and disturbances by drag-and-drop techniques and observe the changes in the oil level. The control panel has buttons and auxiliary windows to set up different options of the simulation: buttons to play/pause/reset, scopes with the most important signals, type of valves, process parameters, timing options, control modes, and disturbances.


First order process simulation: Liquid level

Time (min): 585 | Level (m): 1,6 | U (%): 100,00

Play | Pause | Reset

[✓] Signals
[✓] Type of valve
[ ] Process param
[ ] Simulation
[ ] Manual control
[✓] On/off control
[ ] PI control
[ ] Disturbance

Fig. 6 View of the first-order lag process

It is necessary to mention that every AutoTech's learning resource must be in English and in the mother tongues of the project participants (English, Spanish, Norwegian, German, and Romanian). Thus labs described in this paper were developed taking into account this requirement. When a user gains access to the AutoTech portal and chooses the language, the remote labs and simulators make use of the *Ejs* internationalization features to change the text labels and messages in the view.

5. EXERCISES SUGGESTED FOR WEB-BASED LABS

The exercises suggested for these resources are grouped into three general categories:

- Knowledge of the process. The objective of this kind of exercise is to familiarize the user with the characteristic of the process. For example, to understand the concept of time constant or the steady-state gain.
- Manual control. In these exercises, the student practices with the process using the controller in manual mode. Two types of exercises are:
    (a) *Control on your own*. The user has to change the set-point and/or introduce some disturbance and counteract these actions.
    (b) *Which control action should be used*? The user is requested to analyze the behavior of the

process to determine what action type (direct or inverse) corresponds to the controller.

- Automatic control. These exercises force the student to change the controller parameters according to the background acquired in the previous experiments. Activities suggested:
  - (a) *Understanding the servo and regulation actions*. The experiment is started with the controller in automatic mode, and the user is advised to make changes to the set point or in the output (keeping the set point constant) and observe whether the controller is able to make the process variables reach the new reference.
  - (b) *Basic actions with the PID controller*. These exercises focus on the basic control actions: P, I, and D. Initially, the controller is tuned and running in automatic mode and with P, I or PI control. The student is asked to make a change in the variables and observe whether the controller is able to counteract these disturbances with its current parameters.
  - (c) *Tuning the controller parameters*. The controller is pre-programmed with somewhat inappropriate parameters, and the user is requested to introduce changes into the parameters and observe the system behavior.
  - (d) *Analysis of actuator limitations*. The student makes changes to the parameters and observes whether there is saturation in the control signals. The goal is to understand that certain control objectives cannot be obtained in the real world due to the physical limitations of the process components.
  - (e) *Study of the operating range*. This group of exercises includes experiments similar to those in the previous category. In this case, the goal is for the student to discover the operating range of any plant and to recognize that its control system is always limited.

6. CONCLUSIONS

At university level, the development of virtual and remote labs for control engineering education can be considered a mature technology. Yet, to date, the application of these learning components to non-university education has not appeared in control education journals. It has meant that a wide and significant group of the control engineering community, automation technicians, has been ignored. Thus, a primary target of the AutoTech project is to develop specific advanced learning resources to fulfil the requirements and expectations of these people.

After testing these first versions of the web-based labs described, the feedback received from automation technicians' teachers was very positive. Throughout this year, training pilots are planned to test, assess and validate all the training packages with automation students in order to obtain a final version of these learning resources.

Again, *Ejs* has been shown to be a suitable tool for developing interactive virtual/remote labs. Also, the latest version now includes instrumentation and control symbols in the graphical library. Thus the developer can easily make up virtual and remote labs with views that match the standards of the industrial process control world. A complete English manual on the software tool, Easy Java Simulations, can be downloaded for free from *Ejs*' web server at *http://fem.um.es*.

ACKNOWLEGMENTS

The authors are grateful to the European Commission for support through the Leonardo da Vinci programme, contract N/04/B/PP/165.011 AutoTech, and to the Spanish Ministry of Research through Grant DPI2004-01804.

REFERENCES

AutoTech (2006). http://www.pidstop.com/

Esquembre, E. (2004). Easy Java Simulations: A software tool to create scientific simulations in Java, *Comp. Phys. Comm*, **156**, 199-204.

Sánchez, J., F. Esquembre, C. Martín, S. Dormido, S. Dormido-Canto, R.D. Canto and R. Pastor (2005). Easy Java Simulations: An Open-Source Tool to Develop Interactive Virtual Laboratories Using Matlab/Simulink, *International Journal of Engineering Education*, **21**, nº 5, 798-813.

Buccieri, D.; J. Sánchez; S. Dormido; P. Mullhaupt and D. Bonvin (2005). Interactive 3D Simulation of Flat Systems: The SpiderCrane as a Case Study. 44th IEEE CDC/ECC. Sevilla (Spain).

Dormido, S.; Esquembre, F.; Farias, G. and Sánchez, J. (2005). Adding interactivity to existing Simulink models using Easy Java Simulations, 44th *IEEE CDC/ECC*. Sevilla (Spain).

Duro, N., R. Dormido, H. Vargas, S. Dormido, J. Sánchez and R. Pastor (2005). The Three-Tank System: A Remote and Virtual Control Laboratory using Easy Java Simulations, 44th *IEEE CDC/ECC*. Sevilla (Spain).

Pastor, R., J. Sánchez and S. Dormido (2005). Web-Based Virtual Lab and Remote Experimentation Using Easy Java Simulations. 16th IFAC World Congress. Prague (Cezch Republic).