Laboratories in Engineering Courses via Internet

Mamane Moustapha DODO AMADOU
École de technologie supérieure, 1100 Notre-Dame St. West, Montreal (Quebec) Canada H3C 1K3, dodoamad@yahoo.com

Hamadou SALIAH-HASSANE
Télé-université, 4750 Henri-Julien Avenue, Suite 100, Montreal (Quebec) Canada H2T 3E4, saliah@teluq.uquebec.ca

Maarouf SAAD
École de technologie supérieure, 1100 Notre-Dame St. West, Montreal (Quebec) Canada H3C 1K3, msaad@ele.etsmtl.ca

KEYWORDS: *e-Learning, Simulation, Control System, Remote Laboratory, Virtual Laboratory, Tele-operation*

ABSTRACT: *This paper first presents an approach that makes it possible to create automatic control simulation applications with flexible interfaces. This approach provides the trainer with facilities to generate learner applications according to a given teaching scenario. In order to illustrate the proposed approach, we present an application that allows simulation of a physical process. This application can be used during the laboratory session if adequate material for real experimentation is not available. On the other hand, if adequate material is available, the application can be used for the pre-laboratory period. In the second part of the paper, we present an application that allows remote access via a FieldPoint system to a Virtual Laboratory for controlling the temperature and/or the level of a water tank of a hydraulic workbench. LabVIEW software was used to run these applications.*

1 INTRODUCTION

In universities offering engineering courses, students carry out real experiments in order to verify certain theoretical concepts. Consequently, running real experiments is of major importance for engineering students. However, running real experiments does present some disadvantages. The required equipment can be expensive, the number of tests is often limited in order to prevent premature damage of the equipment and/or to limit the consumed energy by the latter, visualizing some phenomena is impossible because this can lead to the destruction of equipment, etc. In order to overcome these disadvantages, in the last few years, some universities have proposed learning applications that simulate the behavior of studied physical processes.

If the university isn't equipped with the material needed for real experiments, the laboratory session would be limited to simulation. On the other hand, if this material is available, these applications are used for the pre-laboratory period in order to better prepare students for real experiments. These applications can become adequate teaching tools if they have an interface for the trainer and another for the learner. Under these conditions, one must develop applications with flexible interfaces.

But currently, virtual laboratories are now proposed thanks to Internet accessibility [1, 2, 3, 4, 5]. These laboratories make it possible to carry out remote experiments with real equipment, i.e. without being physically on site where the equipment is installed, hence providing several universities with the opportunity of using the same equipment.

This paper presents first an example of an Automatic Control simulation application that has a flexible interface, and secondly an application that allows remote access via a FieldPoint system to a Virtual Laboratory for controlling the temperature and/or the level of a water tank of an hydraulic workbench. The LabVIEW software was used to run these applications.

2 DESCRIPTION OF THE APPLICATION INTERFACE SIMULATING THE PROCESS

An application user interface generally includes commands where the user introduces the necessary data (input parameters) for operating the application, along with indicators where the user can visualize the results (output parameters). The user interface of the applications simulating the real process also has this structure. On the other hand, very often it is necessary to present some buttons, like Start, Stop, etc. in order to better control the simulation.

Let us note Xin, a structure representing the input parameters and Xout, a structure representing the output parameters of the application.

In general,

$$X_{IN} = \{X_s, X_p, X_c\}, \quad (1)$$

where X_s – means structure representing the simulation parameters,
X_p – means structure representing the process parameters,
X_c – means structure representing the command parameters.

The components of X_s can be the simulation time, sampling interval, etc. The components of X_p depend on the objectives of the laboratory. In the simplest case, these components represent the parameters of the process mathematical model. Generally, the transfer function (TF) is used as mathematical model. A TF for linear models has the following general form:

$$G(s) = \frac{N(s)}{D(s)} e^{-Tds}, \quad (2)$$

where s – means Laplace operator,
$N(s)$ – means the numerator of the TF,
$D(s)$ – means denominator of the TF,
Td – means time delay process.

In this case, the components of X_p are the coefficients of the numerator $N(s)$, the coefficients of the denominator $D(s)$ and the time delay Td. The components of X_c can be the test signal parameters (the signals most frequently used are Step and the sinusoidal signal), the controller gains, set point value (the desired value of the process variable, like level, temperature), etc. The components of X_{out} can be the process variables (level, temperature), the control value (opening of the valve, voltage, current).

3 CREATING TRAINER AND LEARNER APPLICATIONS

The proposed approach consists in providing a trainer application that has an interface where all the components of X_{IN} and X_{OUT} structures are accessible. Starting from this application, the trainer must be able to easily generate learner applications needed for a given teaching scenario. Let us note that for certain scenarios, as we will see below, one must hide certain X_{IN} and/or X_{OUT} components, since these components represent the parameters that learners must determine by using a given technique. Consequently, learners must not have access to the application program.

In order to implement the proposed approach, LabVIEW of National Instruments has been used because it offers a simple method of programming and the needed functionalities to achieve our goals. Recall that LabVIEW applications are called virtual instruments (VI) [6]. A VI consists of a front panel (user interface) and a block diagram that represents the application program. The controls (where the user introduces the data) and the indicators (where the user displays the results) constitute the objects of the front panel. Each object of the front panel has its corresponding terminal on the diagram. In our case, the components of X_{IN} and X_{OUT} structures constitute the objects of the front panel of the VI.

LabVIEW has functionalities that allow one to show or hide the objects of the front panel either from the front panel, or from the block diagram. By using these functionalities, the trainer can generate

LabVIEW applications with interfaces where some components (according to the considered teaching scenario) of X_{IN} and X_{OUT} structures will be hidden. But such a VI should not be provided to learners because they will have access to the program. In this case:

- Learners can occasionally modify the application program and consequently change the tasks configured beforehand by the trainer,
- Learners will be able to show the components of X_{IN} and X_{OUT} structures hidden beforehand by the trainer by using the functionalities of LabVIEW.

To make the program inaccessible to learners, the trainer must transfer the LabVIEW learner applications to executable versions. These have the same interface as their corresponding LabVIEW applications, but do not contain a block diagram (program). To create executable versions, LabVIEW has an additional module, Application Builder, that allows one to easily create executable versions. Another equally important advantage is that these executable versions can be run on computers that do not have a LabVIEW license. In other words, learners are not obliged to have the LabVIEW software installed on their computers. They will simply have to download a free LabVIEW Run-time Engine Software.

In the following section, we will see a specific application to illustrate using the proposed approach.

4 PRACTICAL APPLICATION FOR SIMULATION

One of the laboratories taught to engineering students deals with automatic control systems, such as level control, speed control of a DC motor, etc. These laboratories generally contain three parts: identification, calculation of the controller gains according to a given technique and controller validation. Process identification consists in determining its mathematical model (structure of this model and corresponding parameters) based on experimental input and output observations. The parameters of the model determined at this stage will be used to calculate the controller gains.

To implement this kind of laboratory via simulation, we needed to create an application. The created application interface is made up of three pages. We named these pages **Process**, **Identification** and **Regulation**. To view one of these pages, simply click on the corresponding tab. Each page is documented in the following way. A window, Context Help, contains information about each objet of the interface on the one hand, and on the other hand, each interface has a description.

As previously noted, in simulation applications, the parameters of the process mathematical model constitute the components of the X_{IN} structure. This is necessary because in this case, the trainer can choose the suitable model for experimentation. To imitate real experimentation, the parameters of the mathematical model of the studied process must be hidden for learners. As during real experimentation, learners must use the test signal and the output response to identify the process. The test signal and the output response will have to be available on the interface of their application.

In our case, as mathematical model, we chose the TF in order to be able to simulate the behavior of various processes. The **Process** page of the trainer interface of our application is shown in Figure 1.

On this page, simulation parameters and process parameters are shown. The trainer uses this page to introduce simulation parameters and the mathematical model of the studied process.

Once these parameters are introduced, the trainer uses the functionalities of LabVIEW to hide process parameters. This procedure takes just a few clicks.

The other pages of the interface of the application, i.e. the **Identification** and **Regulation** pages, are presented in Figures 2 and 3 respectively. To achieve experimentation, learners will need all the objects that are on these pages; consequently, the learner application interface will have the same pages.

As one can see, to obtain the learner interface from the trainer interface, one simply has to click a few times with the mouse. Once the learner LabVIEW application is generated, the trainer uses the module Application Builder to create the executable versions. The application generated by the trainer in the form of an executable version can be used by learners either during a pre-laboratory session or during the laboratory session if the required physical setup is not available for real experimentation.

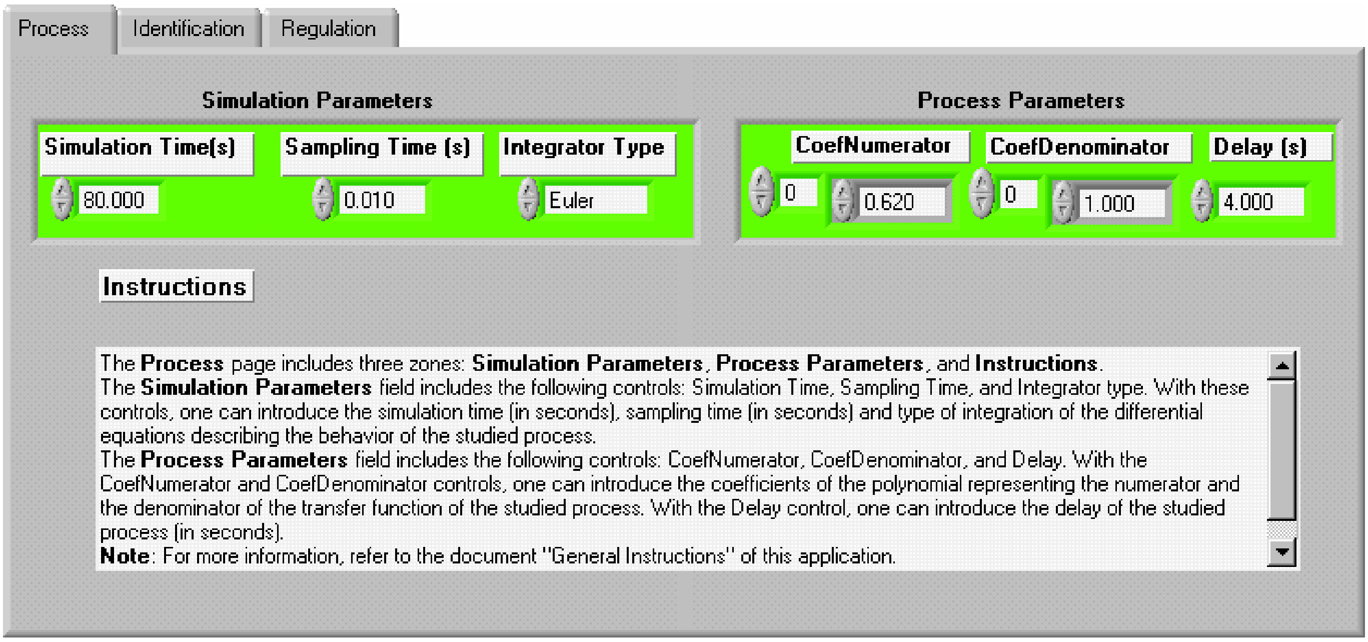

Figure1 - **Process** Page (Trainer)

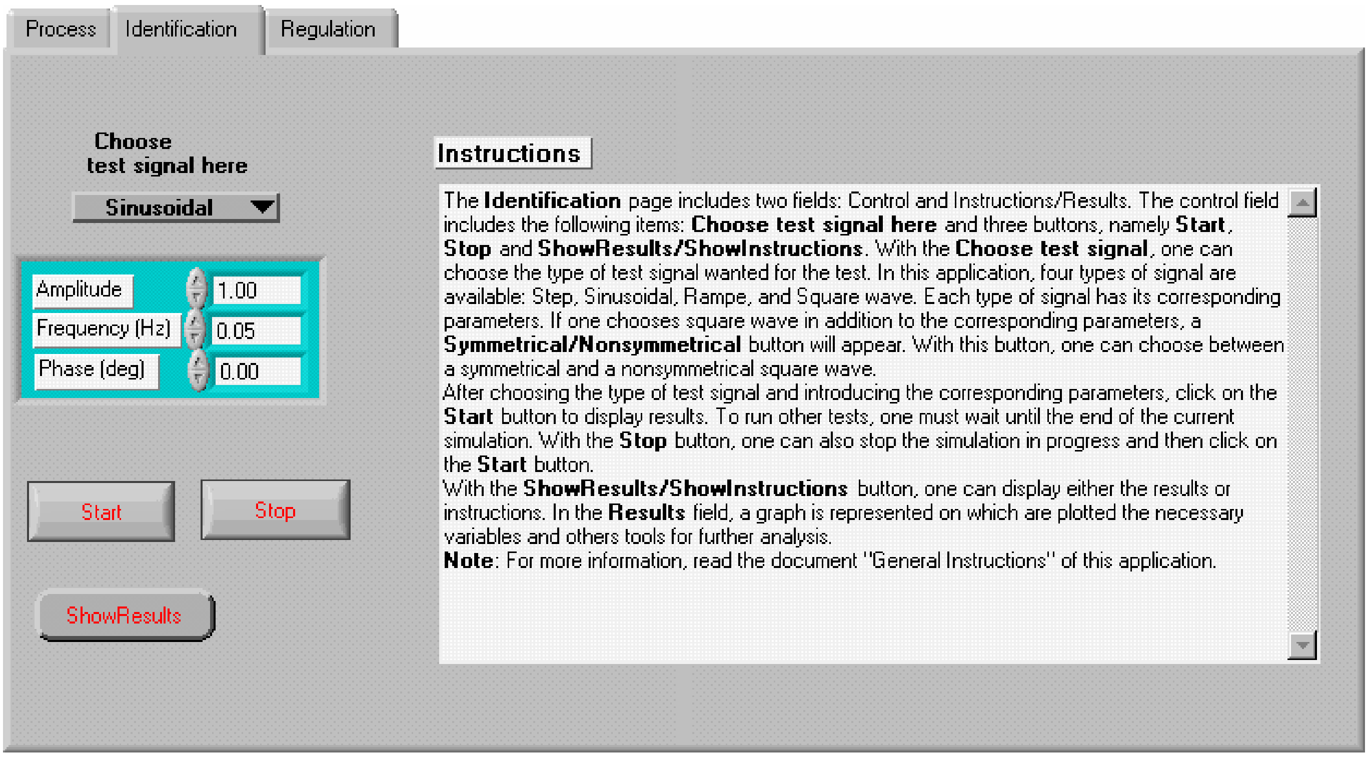

Figure 2 - **Identification** Page (Trainer and Learner)

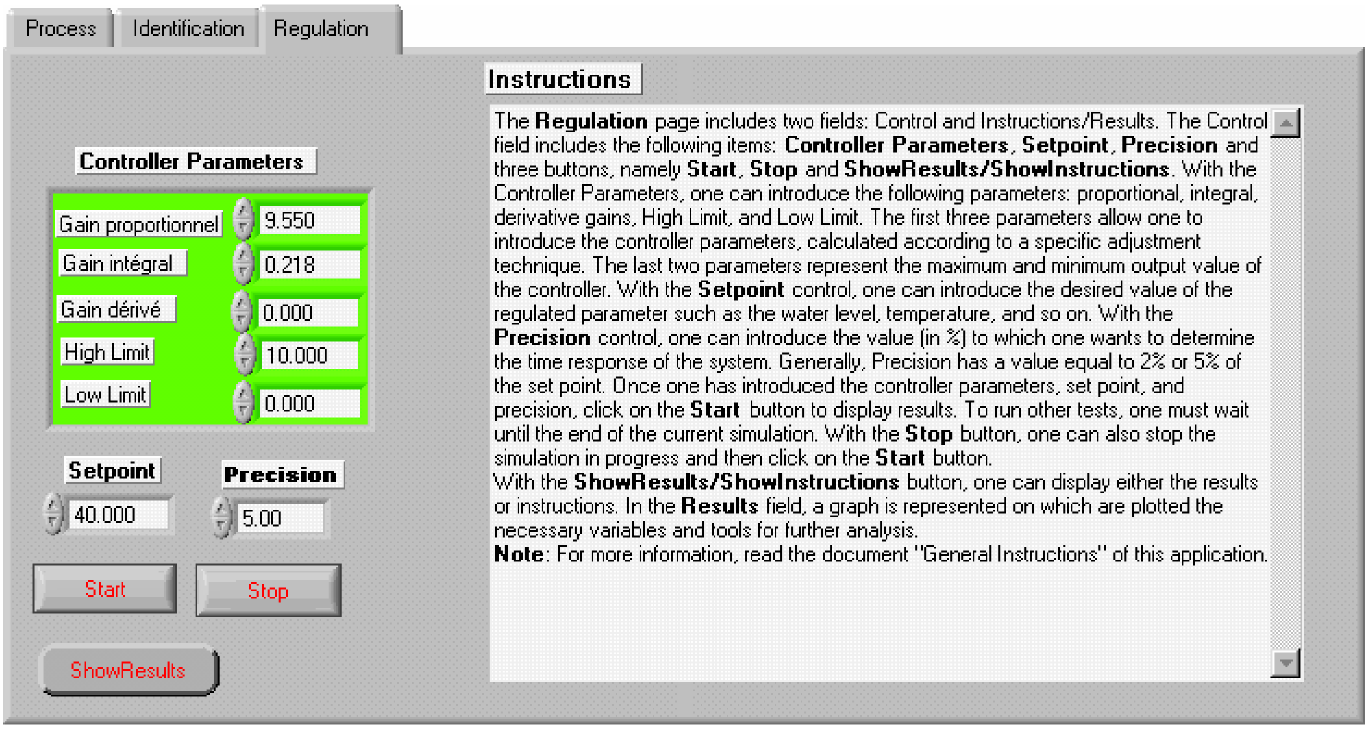

Figure 3 - **Regulation** Page (Trainer and Learner).

In the rest of this article, we will present an example of remote access via a FieldPoint system to a virtual laboratory for controlling the temperature and/or the level of a water tank of a hydraulic workbench. This hydraulic workbench can be used to perform real experiments, either locally or remotely, after performing simulation with the above-described application.

5 TEMPERATURE CONTROL AND/OR LEVEL CONTROL OF AN HYDRAULIC WORKBENCH VIA AN INTERNET NETWORK

5.1 DESCRIPTION OF THE HYDRAULIC WORKBENCH

The workbench allows one to control the level and/or the temperature of a water tank. Two servovalves are used to control the input flow of cold and hot water. The outgoing flow is controlled by a third servovalve. The system is also equipped with one pressure transmitter (600T series), and one temperature sensor.

The three servovalves are controlled automatically by the software. FieldPoint system, a product of National Instruments, is used to control these valves, and to acquire the water level and/or temperature.

Generally, a FieldPoint system consists of a network interface module, one or more terminal bases, and one or more I/O modules. The terminal base is connected to the network interface module. The I/O modules are mounted on the terminal base. The I/O modules interface with the different sensors and actuators of the physical setup.

In our case, the FieldPoint system consists of a network interface module of type **FP-1600**. This interface module has a power supply and an input that is directly connected to an Internet network via a fixed IP address.

One terminal base of type **FP-TB-10** that can hold up to six dual-channel I/O modules.

The I/O modules we are using are:

- One analog current Input module (4-20mA) of type **FP-AI-C420**. One channel of this module is used for data acquisition of the level and the second, for the temperature.
- One analog current Output module (4-20mA) of type **FP-AI-C420**. One channel of this module is used to control the servovalve of the cold input flow and the second, the servovalve of the hot input flow.

- One analog voltage Output module (0-10V) of type **FP-AO-V10**. One channel of this module is used to control (via a voltage to current converter) the servovalve of the outgoing flow of the tank.

5.2 ACCESS TO THE HYDRAULIC WORKBENCH

The above-described bench can be accessed remotely via an Internet network from a host or client computer. The host computer has the following software tools: the "Measurement & Automation Explorer" (**MAX**) or "Fieldpoint Explorer" that is used to detect and to configure the Fieldpoint hardware, along with the LabVIEW application that manages the physical setup. In the client computer, the free LabVIEW run-time engine must be installed. Both host and client computers must be connected to the Internet Network.

The particularity of using the FieldPoint system is that both the host and client computers may be at any geographical location. In our case, the former is in the same room as the FieldPoint system and the hydraulic bench. This allows us to use a camera and microphone in order to provide visual and audio feedback to the user.

In the following procedure, we assume that the trainer has access to the host computer, and that the learner has access to the client computer.

After detecting and configuring the Fieldpoint hardware with **MAX**, the trainer can launch the LabVIEW application to control the workbench.

The LabVIEW "Web publishing Tool**"** allows, without any additional programming, remote control this application through a common Web browser. Refer to [5] for more details about enabling remote panels, client operation, application administration, and application security with the "Web publishing Tool".

So, in order to give the learner remote access to the physical setup, the trainer must perform the appropriate configuration in the host computer using the "Web publishing Tool**"**. After completing the appropriate configuration, the trainer will get the URL address of the LabVIEW application that manages the bench. Using this URL address (provided by the trainer), the learner from a Web navigator will be able to see the front panel (without block diagram) of the above-mentioned LabVIEW application on the screen of his/her computer. From this "imported" front panel, the learner can control the LabVIEW application located in the memory of the host computer, and consequently the workbench can be manipulated remotely via the Internet network.

The "imported" front panel on the client computer screen is shown in Figure 4 and the front panel of the host computer LabVIEW application is shown in Figure 5. These figures show a PID controller application for regulating the water level in the tank of our physical setup. One notes that the results obtained are identical.

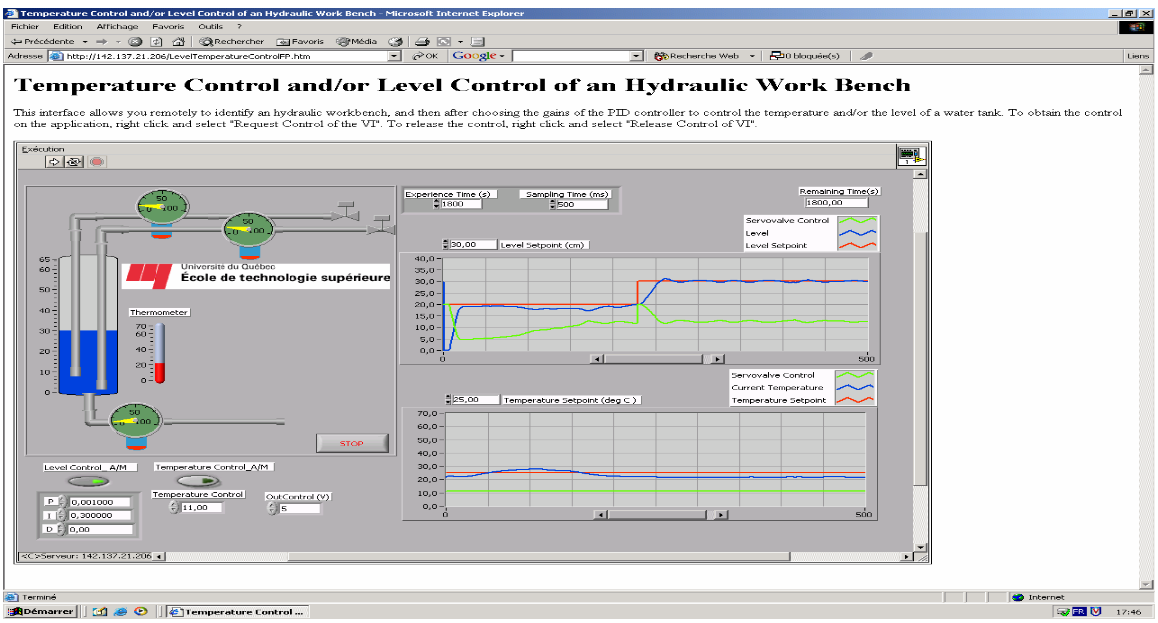

Figure 4 - "Imported" Interface of the VI Managing the Workbench (Client Computer)

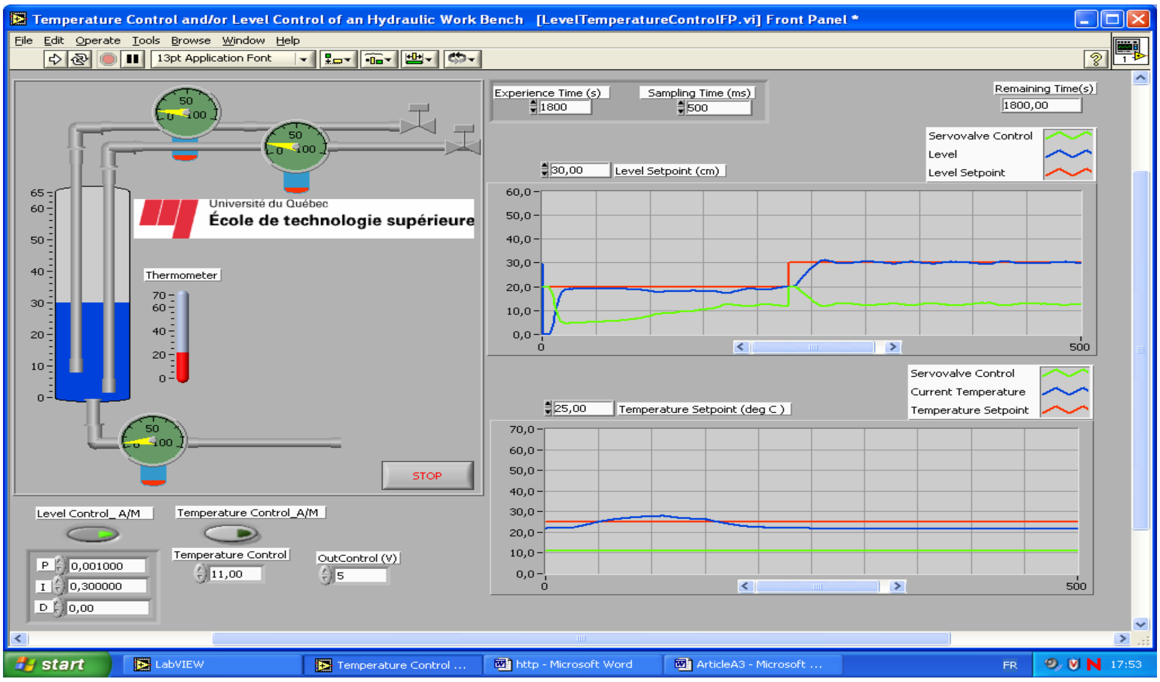

Figure 5 - Interface of the VI Managing the Workbench (Host Computer)

6 CONCLUSIONS

This paper proposes an approach in which the trainer can generate learner applications to simulate industrial processes according to a given teaching scenario. This approach can be used to create similar applications used during pre-laboratory and laboratory sessions. These applications will help learners to better understand certain physical phenomena.

LabVIEW is an attractive environment for creating simulation applications for trainers, learners and virtual laboratories that are accessible either locally or remotely.

REFERENCES

[1] Saliah, H., Saad, M., De La Teja, C., “Virtually and Remotely Accessing and Controlling Laboratory Real Devices: A New Trend in Teaching and Learning in Engineering,” ITHET2000, Istanbul, July3-5, 2000.

[2] Maarouf Saad, Hamadou Saliah Hassane, Hussein Hassan, Zouhair El-Guetioui and Mohamad Cheriet, “A Synchronous Remote Accessing Control Laboratory on the Internet, International Conference on Engineering Education,” August 6-10, 2001 Oslo, Norway.

[3] Ingvar Gustavsson, “Laboratory Experiments in Distance Learning, International Conference on Engineering Education,” August 6-10, 2001 Oslo, Norway.

[4] Kevin Amaratunga, Raghunathan Sudarshan. “A Virtual Laboratory for Real-Time Monitoring of Civil Engineering Infrastructure,” International Conference on Engineering Education, August 18-21, 2002, Manchester, U.K.

[5] Gillet D., Ch. Salzmann, R. Longchamp, and D. Bonvin, “Telepresence: An Opportunity to Develop Practical Experimentation in Automatic Control Education,” European Control Conference, Brussels, Belgium, 1997.

[6] LabVIEW User Manual, http://digital.ni.com/manuals.nsf/websearch/790127B60590AD0C86256D2C005DCE0F?OpenDocument&node=132100_US

[7] Distance-Learning Remote Laboratories using LabVIEW, http://zone.ni.com/devzone/conceptd.nsf/webmain/7BD0B01FCF3CF61A86256B510059F0FB/$File/WP2238.pdf