REAL TIME CONTROLLED LABORATORY PLANT FOR CONTROL EDUCATION

Roberto Saco[1], Eduardo Pires [2] and Carlos Godfrid[3].

Abstract — *This paper describes an experimental installation for performing several basic experiments in Automatic Control courses. The process to be controlled consists of three interconnected water tanks. It is possible to use just one tank, or two or the three of them adjusting the corresponding valves. The controller is based on a dSpace board that implements a real time interface. The control strategy is described in block diagrams in Matlab/Simulink, software that students are familiar with. They use this environment for the modeling, analysis and design stages of their control projects. With these software tools they can also configure the actual controller, implementing their designs using Simulink blocks. As they don't need to do any special programming or debugging, they can focus on the control items. The learning process becomes more effective if they can use the same well-known environment to implement the real controller.*

Index Terms — *control education, educational equipment, hardware in the loop simulation, real time control.*

INTRODUCTION

Simulation tools are frequently used as an educational aid in automatic control courses. Initially, analog computers with electronic circuits were used for simulating different types of physical processes. When digital computers appeared, several simulation software packages based on numerical techniques were presented. Personal computers, low priced and universally disseminated, contributed to incorporate simulation tools as a standard educational resource in systems and control education.

It is usually recognized that, studying and experimenting with simulation models, students can experiment with systems that are impossible to work with in a laboratory, being potentially dangerous or huge dimensions or very expensive processes. In this way, it is possible to obtain some knowledge about its modeling, operation or stability aspects without being in actual contact with the plant. But, at the same time, this is the major drawback of this approach: being in contact with the real process and its associated hardware is an important issue concerning engineering education. Even if it were possible to have a laboratory process, to design and build the electronic circuitry required for its measurement and control would require a lot of time, more than the available in a regular semester course. A different approach is possible using real time simulation tools, and it is also possible to develop laboratory applications in short time, using a PC and low cost hardware.

MODELING AND REAL TIME SIMULATION

Modeling a system and writing the simulation program contributes to a better understanding of its physical principles and properties. Defining the equations of the different parts of the system and how they interact with each other, students can obtain a clear understanding about the system structure and the way it operates. In addition, modeling the controller shows how control forces are generated and how they interact with the process. If, additionally, these models can de simulated in “real time” students get a clear idea in regard to values of physical variables and typical response times.

From an educational point of view, learning is improved when a physical plant is available for modeling and experimenting. In this case, students can compare their simulation with experimental data. They can also design control algorithms based on the mathematical model, simulate how the controlled system works and then implement the controller of the physical process. But, even if conceptually simple, it is not easy to put this approach into practice in the short time of a regular academic course. Typically, a control project involves several steps, when performed with usual development methods:

- Defining the electronic device for implementing the controller. It could be a PC, a microcontroller, or a DSP based board. This device receives sampled signals, computes the control algorithm, and generates the control signal
- Electronic hardware design of the data acquisition subsystem: D.A.C. , A.D.C., filters.
- Writing code for the control algorithm and the software required for real time tasks (interrupts, priority management, control algorithm execution)

Recently, software tools for real time control became available. They include most part of the former steps, so the design process is greatly simplified. Using this software is

[1] Roberto Saco, Universidad de Buenos Aires, Facultad de Ingeniería, rsaco@fi.uba.ar
[2] Eduardo Pires, Universidad de la Patagonia San Juan Bosco, epires@satlink.com.ar
[3] Carlos Godfrid, Universidad de Buenos Aires, Facultad de Ingeniería, Departamento de Electrónica, carlos.godfrid@ieee.org

possible to output values while the simulation program is running, and also to add signals obtained from external sensors. This scheme is known as "hardware in the loop" simulation. Control and supervisory strategies are designed graphically in the Simulink block diagram environment. Then, control algorithms are downloaded to a real-time prototyping system, instead of designing specific hardware.

HARDWARE AND SOFTWARE

Software of the Matlab environment is used in these real time experiments. This is a well known environment, as students use Matlab and its toolboxes in several courses during their first academic years

SIMULINK.

In a broad outline, Simulink is a software tool for modeling, analyzing and simulating dynamic systems. It supports linear or nonlinear models, and continuous, discrete or hybrid time systems. Simulink offers a Graphic User Interface (GUI) that facilitates the model definition as a block diagram in the usual way control engineers do. So, it is not necessary to write a computational model, coding the difference or differential equations in some programming language. Simulink offers a set of library blocks, some of them in the main program and others grouped in "toolboxes" concerning a specific subject. Models defined in this way can be simulated using different integration methods and results can be obtained in numerical form for data processing or presented in a graphic display.

MATLAB RTW.

It is possible to obtain C code representing the Simulink model, using the Real Time Workshop, [1]. It is also possible to generate a complete C program that can be executed in a real time platform, independently of Matlab or Simulink. In this way, continuous or discrete time systems can be executed in different types of computing devices. As these systems may include input/output hardware, it is easy to build from simple controllers to others including complex algorithms. Generation of a C program with RTW is an automated process, and students are relieved from cumbersome hand coding. The same approach used for controller design can be used for real time simulation of processes, where the input/output signals obtained, simulate the physical variables of a real plant.

HARDWARE.

There are several hardware options when working in the Matlab Real Time environment. In our experiments we used two of them. In the first of them, we used a standard National Instruments LAB PC 1200 board. This board has 8 analog 12-bits input channels, 2 analog output channels and 24 digital input/output channels. Drivers for this kind of board is included in the RTW toolbox. Using this hardware, control experiments as the stabilization of an inverted pendulum were done. In this application, the PWM driver for the electric motor and the state variable feedback algorithm were defined as block diagrams in Simulink. In the other approach, used in the prototype plant presented in this paper, a dSpace 1102 floating-point controller board was used [2]. This is a single board system, designed for implementing digital controllers and real-time simulations. It is based on the Texas TMS 320C31 floating point DSP and includes on-board A/D and D/A conversion as well as digital I/O and incremental sensor interface. When using the RTW with the dS1102 board, target specific software is developed using dSpace Real Time Interface to Simulink. Matlab RTW and dSpace RTI provide an integrated environment that automates the tasks for building an executable that is downloaded onto the dSpace processor board.

THE LABORATORY PLANT

Liquid level regulation systems are classical experiments in control labs. The way the system operates is very clear. Students can intuitively appreciate results of increasing or diminishing controller gains. It is very easy to introduce known and measurable disturbances. Specifications and physical magnitudes become meaningful, and they can appreciate if control objectives are achieved. In our case, we have projected a three tank prototype process, with a pneumatic valve actuator for water flow control

An experimental process for doing the general experiments required in a basic controls course was our main objective. We conceived it to be useful for doing more complex experiments and also to be used in advanced courses. It is possible to use just one tank, or two or the three of them adjusting the corresponding valves, as shown in Figure 1.

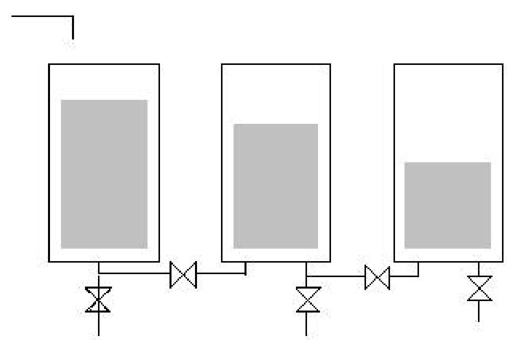

FIGURE. 1
DIAGRAM OF THE THREE TANK SYSTEM

An electrical current generated by a D.A.C. in a PC board, controls the pneumatic valve opening. Liquid level is measured with pressure sensors in each tank linked with A.D.C.´s in the PC. As mentioned, a DSP based board, where real real time control algorithms are running, connects the PC with the level sensors and the pneumatic valve.

TANK MODELLING ISSUES

A first lab experiment consists of controlling just one tank. Students measure and calculate physical parameters to build the required mathematical model and the corresponding Simulink diagram. Then, they can design control systems and analyze its operation and performance running a Matlab simulation. A typical example of a first control course is to state and linearize the equations of a tank, (Figure 2)

$$q_i(t) - q_o(t) + q_d(t) = A\frac{dh(t)}{dt} \qquad (1)$$

An approximate expression for the outgoing flow is:

$$qo(t) = \frac{h(t)}{Ro} \qquad (2)$$

When trying to evaluate R_0 , students understand at a glance that it is not possible to find a constant value for R_0. Instead, they need to perfect their model, showing the nonlinear relation between height and output flow. A block diagram that takes account of this fact is shown.

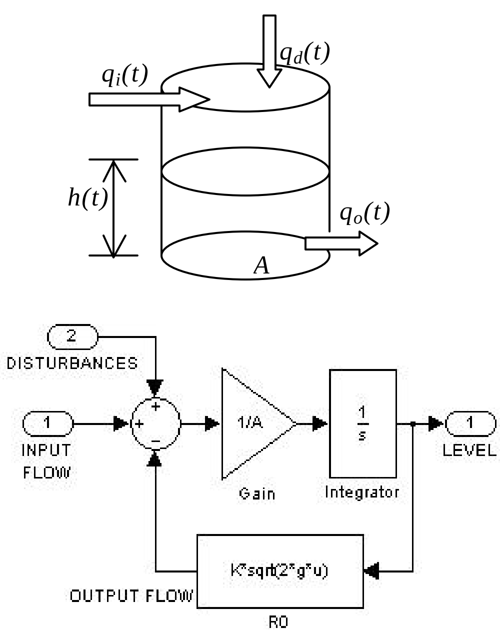

FIGURE. 2
PHYSICAL VARIABLES AND BLOCK DIAGRAM OF A SINGLE TANK

Measuring flow and level in several operating points, it is possible to draw a curve. Using Matlab tools, it is possible to approximate the experimental data with a mathematical equation, as required to complete the block diagram. Figure 3 depicts field data and two approximating formulae.

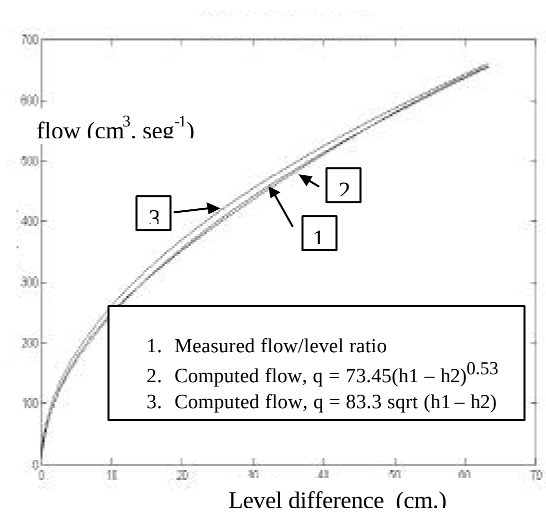

FIGURE. 3
PHYSICAL VARIABLES AND BLOCK DIAGRAM OF A SINGLE TANK

The model achieved still doesn't reflect properly how the prototype plant operates. Overflow, for example, is not considered in this model. It doesn't consider that there could be external pressure opposing output flow either. Taking into account these and other specific details observed in the actual process, students modify their diagram until simulation results and physical measurements agree. Using software resources of the Matlab environment, dynamic equations, non-linearity and specific details can be grouped in an S-function. This S-function can be represented in a block form, as shown in Figure 4 and used in a Simulink block diagram.

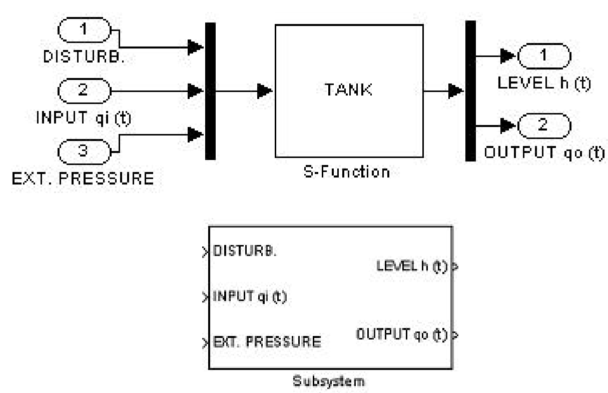

FIGURE.4
SIMULINK BLOCK OF A SINGLE TANK

This approach allows connecting blocks in a diagram relating variables in a similar way as physical variables do. Figure 5 shows an example where the first tank receives a constant flow input while the second receives the output of the first one and disturbances.

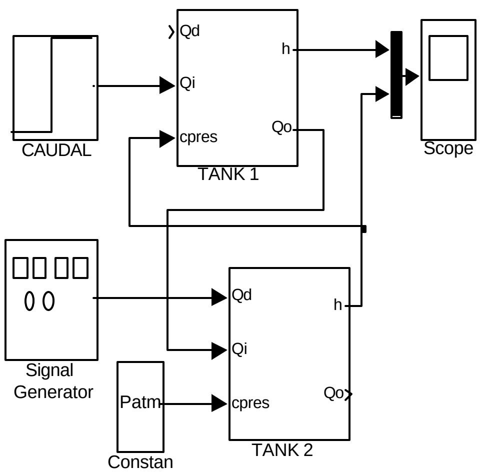

FIGURE. 5
SIMULINK DIAGRAM OF TWO INTERCONNECTED TANKS

CONTROLLER MODELING ISSUES

A first set of experiments is related to controlling just one tank. Students experiment with proportional, proportional-integral and P.I.D. controllers. They design the control law as a Simulink block diagram, they tune controller gains and analyze how system performs, simulating the controlled process. Finally, they can verify their designs working over the real plant. Using Matlab Real Time Workshop the same controller used for simulation is connected, via A.D.C.´s and D.A.C.´s to the physical process. A photograph of the experimental installation is shown in figure 6.

It is possible to experiment with more advanced control schemes. As the prototype plant measures three physical variables, the three tank levels, a State Variable Feedback approach can be implemented. Also, it is not difficult to include a linear observer as part of the control strategy, [3].

Using additional support tools, as TRACE and COCKPIT from the dSpace environment, it is possible to view signals, collect data in a Matlab ".mat file", and change parameters while the model runs on the target processor.

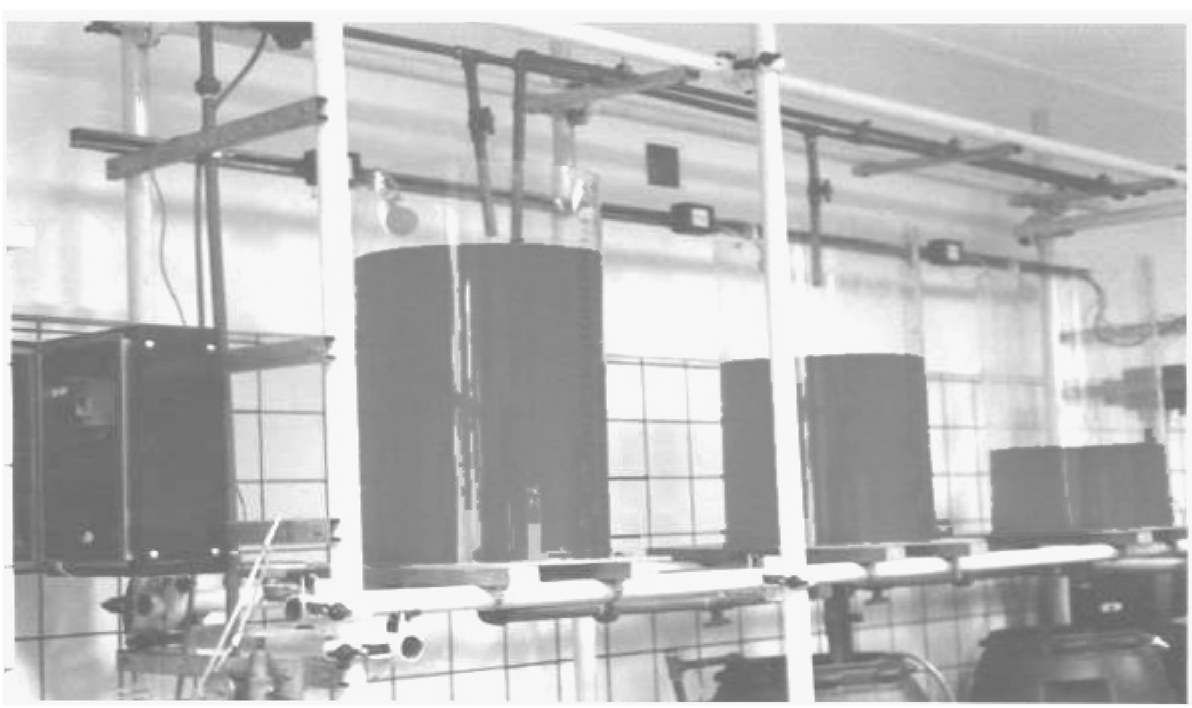

FIGURE. 6
PHOTOGRAPH OF THE THREE TANKS SETUP

For analysis pourposes, measurements can be displayed in one or multiple plot windows, presented as y/t or y/x plots, and transfer data to the Matlab workspace. For control design purposes, students can design their own virtual instruments. They choose from a library that offers them several different output objects (displays, gauges, moving bars...) and different input possibilities for interactive modification of the controller variables (sliders, buttons, numeric input from keyboard and so on).

Figure 7 shows a virtual instrument for an IMC controller, displaying measured and computed liquid level and the analog signal controlling the pneumatic valve.

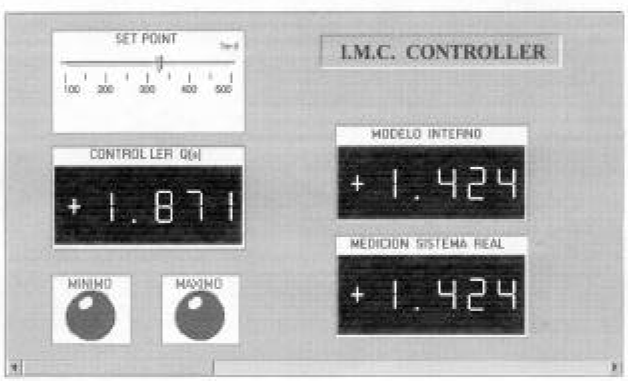

FIGURE. 7
VIRTUAL INSTRUMENT OF AN IMC CONTROLLER

CONCLUSIONS

Frequently, in systems and control courses, we deal with simulated processes instead of real ones that won´t be possible to have in an educational laboratory. If it were possible to have access to a real plant, students would get a better approach to engineering real life problems and learning would became more effective, but design, programming and construction of control circuits would require an excessive amount of time. New real-time software tools facilitate rapid prototyping of control systems. In this paper a simple prototype plant, controlled by a Real-Time Matlab based control station was presented. Working in this environment, students model, analyze and design control strategies using Matlab/Simulink and they apply directly the designed controllers to the physical process.

An innovative aspect of the presented pilot plant is that it is not limited to a fixed set of experiments. It can grow in complexity as students advance in their studies. It is useful for studying elementary properties of closed loop systems. For doing this, it is enough to use just one tank to have a first order system. Connecting more tanks they are gradually faced with problems that are more difficult. Using Matlab/Simulink RTW, it is easy to implement and experiment with different control strategies, improving control engineering education.

ACKNOWLEDGMENT

The installation of the prototype process was supported by the University of Buenos Aires, as part of the Research and Investigation project TI032 “Experiments on new approaches on industrial control”.

REFERENCES

[1] The Mathworks. *“Real Time Workshop for use with Simulink”*, User´s Guide . Natick, MA., USA.

[2] DSpace, “ *RTI 1102 Real Time Intereface to Simulink Users Guide”*, Paderborn, Germany, 1997.

[3] Goodwin, C., Graebe, S., Salgado, M., *“Control System Design”*, Prentice-Hall, New Jersey, USA, 1991, pp. 533,538.

[4] Berber,R, "A model undergraduate control course integrating IMC technology and an experimental system", *ACE ’97 Preprints, IV IFAC Symposium on Advances in Control Education*, July 14-16,1997, Istanbul, Turkey, pp. 277,282

[5] Cistac, G., Godfrid, C, "Protoype Process for Basic Experiments within an I.M.C. approach", *“Proceedings of Combelet ’99, Computer based experiments, learning and teaching”*, Wroclaw, Poland., 1999, pp. 19,25

[6] Morari, M., Zafiriou, E., *“Robust Process Control”*, PTR Prentice Hall, Englewood Cliffs, New Jersey, 1989, pp. 44,45