A JAVA BASED SIMULATOR FOR BASIC CONTROL EDUCATION

D. Limon [1] **F. Salas** [1] **D.R. Ramírez** [1] **T. Alamo** [1]
E.F. Camacho [1]

Abstract: In this paper we present a java based simulator for control education in basic courses. The application has been developed using the well known tool Easy Java Simulation.

The objective of the application is to help the student to learn the design of classic controllers such as P,PI, PID, etc testing the tuning procedures to control the position of an antenna controlled by a DC motor. Thus the application allows the student to choose the parameters of the antenna and the DC motor, to choose the controller to be used and its parameters and finally to simulate the closed loop system observing the evolution of the signals as well as a 3-D view. Furthermore, in order to show the real behavior of the system, dead zone, saturation, disturbances and non-linearities can be added to the model. This application has been used by the authors to teach a basic control course at Escuela Superior de Ingenieros (University of Seville) as virtual laboratory.

Moreover, since the application is java based, this can be used by the students from the authors' web pages and this can also be installed in the student's laptop (whichever the platform is) by downloading it from the authors web page (Limon and Salas, 2003).

1. INTRODUCTION

Computer-Enhanced Learning (CEL) techniques are of great interest in teaching control or more generally engineering topics to under-graduated or graduated students (Heck, 1999; Dormido, 2004). In recent years the engineering programs at most universities in developed countries have change to include CEL techniques as a basic tool in the typical engineering courseware. Interactive simulators allow the students to learn the control systems design methodology interactively by selecting the most adequate control strategy or even developing their own control algorithms. The results of their choices are quickly displayed on the computer screen so they can asses their level of assimilation of control design methods and explore the effect of changing from a given controller to a different one. Moreover, it allows the teacher to concentrate in the conceptual aspects of the design methods preventing the unwilled effect of making control design appear as a mathematical problem without connection to the engineering practice (Dormido *et al.*, 2004).

The increased computational power also allows the use of more realistic models removing the need of excessive simplifications of real systems dynamics. CEL techniques can also be used to establish virtual control laboratories with two main advantages: the student can practice control methods wherever he is with just a computer and also institutions that cannot afford large laboratories can introduce practice to lower grade subjects that are usually crowded with many students.

This paper presents a CEL program designed to help the students to train in the classic design of controllers applied to a motivating example: the control of an antenna controlled by a DC motor. Some of the parameters describing the dynamics of the motor and the antenna can be set so that the system to be controlled is effectively different for each student. This proved to be essential to get a clear view of its understanding of the subject since makes the student to be involved in the solution of his/her own problem.

The implemented controllers are the classic P, PI or PID types as well as lead and/or lag networks. All their parameters can be set to test the closed loop system. The graphical interface of the program quickly displays the results of the simulation as well as a 3-D view of the antenna in motion. Furthermore, certain nonlinearities such as saturation and dead-zones can be added to the system in order to raise the difficulty of finding the correct parameters of the controller. Also, the student can get a more realistic experience of PID controller design.

The program presented here is written using the **Easy Java Simulations (EJS)** package. This tool provides a very convenient way to develop CEL simulation programs for many scientific disciplines. The programming is not as difficult as using a bare programming language. One of the greatest benefits of using this tool is that being it based in java it produces programs that can be run in different hardware platforms, and also they can be used in any computer with an internet browser capable of running java applications. An additional benefit is that the code is royalty-free, thus institutions with tight budgets can save in buying large quantities of licenses of well known mathematical packages.

The paper is organized as follows: first a brief description of the Easy Java Simulations is presented in section 2. In section 3 the problem to be solved by the students is formulated. In the following section the application is described and the paper draws to a close with some conclusions.

2. EASY JAVA SIMULATIONS

Easy Java Simulations is a software tool designed for the creation of computer simulations (Esquembre, 2004). Easy Java Simulations was developed in the context of the Open Source Physics Project at the University of Murcia, Spain (Esquembre, 2002). The software platform, and the simulations created with it, can be used as independent programs under different operating systems, or be distributed via the internet and run within html pages by most popular web browsers. Easy Java Simulations is designed for people who are more interested in the content of the simulation, the simulated phenomenon itself, and much less in the technical aspects needed to build the simulation program. Hence, it provides a conceptual structure and simplified tools that allow designers to concentrate most of their time on the description of the model of the phenomenon they want to be simulated. The typical audience includes science students, teachers and researchers who have a basic knowledge of programming, but who cannot afford the big investment of time needed to create a complete graphical simulation.

Most computer simulations of scientific phenomena can be described in terms of the model-control-view paradigm. This paradigm states that a simulation is composed of the model, which describes the phenomenon under study, the control, which defines the actions to be performed on the simulation and the view, which shows a graphical representation of the different states of the phenomenon through the simulation. To simplify the construction of a simulation, Easy Java Simulations suppresses the control part, merging it half into the view, half into the model. Creating a simulation using this tool consists in defining its model and its view (i.e., the GUI or graphical user interface) and establishing the mutual connections needed for the correct visualization of the state of the phenomenon being simulated and the appropriate interaction of the user with the view (either to modify this state or to perform the actions defined on the model).

This explicit separation in parts reinforces conceptually the central role of the model of a simulation. It is the model that defines what the program simulates and how. There may be different views for a given model. Teachers can create the same simulation with different GUIs for different tasks or different students.

In addition to the Model and View, Easy Java Simulations has one more component from which a simulation is built the Introduction. For pedagogical or scientific purposes, it is always helpful to include a description of what a simulation does, including the instructions on how to operate it and other pertinent information. This information appears in the Introduction, which is used to generate the content of the html web page. Therefore, there are three major parts to the interface: Introduction, Model and View.

3. PROBLEM FORMULATION

The objective of this application is to teach engineering students the design of classic controllers in basic control courses. The student receives the theoretical foundation in the lectures and some numerical problems are solved during the classes. In order to make the student solves a realistic control problem, a prototype problem is proposed: the control of the position of a servomechanism. A different set of physical parameters of the servomechanism is chosen for each student and they must design and test the explained controllers according to some given specifications.

Currently, one of the actual problems in control education is the shortage of money and space to create and maintain laboratories. Due to this, many developments in virtual and remote labs through the Internet have received great attention in the last few years. Making use of computer simulations in an educational context means using the computer to build theoretical models or to model real-world phenomena in order to help students gain insight into the behavior of complex systems. Interacting with an instructional simulation can help students to gain a better understanding of a real system, process, or phenomenon through exploring,
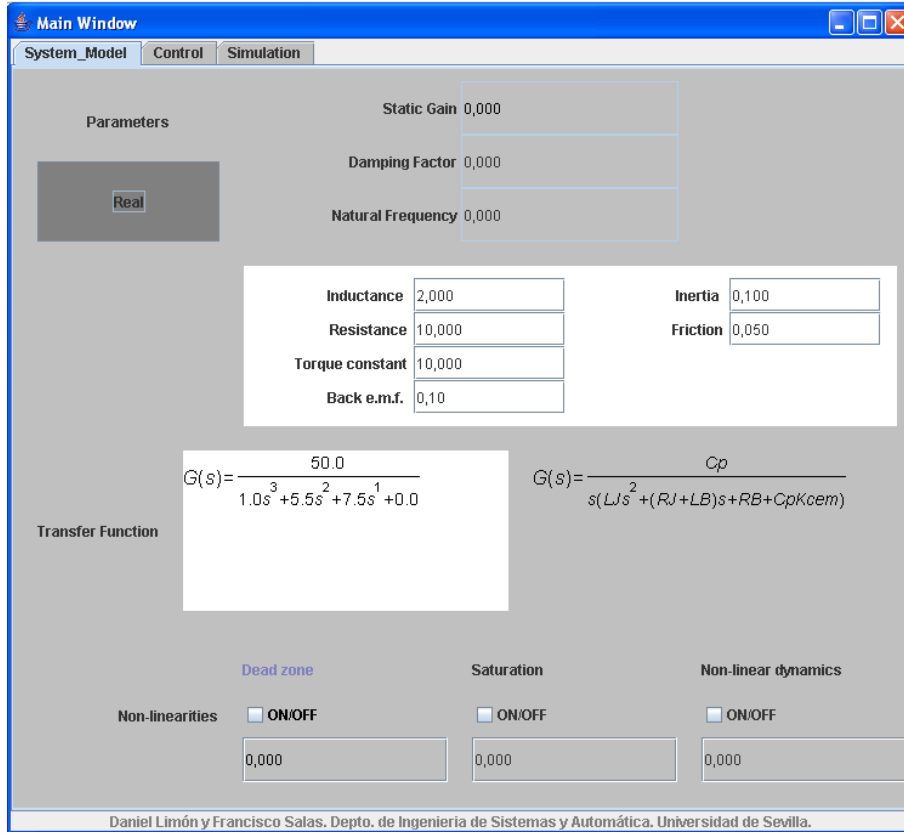
Fig. 1. Main Window: System model panel

testing hypotheses, and discovering explanations for the mechanisms and processes.

Therefore, to test the solution of the proposed servomechanism and experiment the real engineering problem they are dealing with, an interactive simulator has been designed. The different parts of this application will be shown later on. In what follows, a brief description of the problem to solve is given.

The tool tries to show (with a very simple and well known system) the effects of the different parts of the controllers introduced in the theoretical lectures and motivate the students to analyze the effects of undesired perturbations and nonlinearities in the global behavior of a system modeled and controlled by linear (classical) methods.

3.1 System model

This example has a electrical part that models a "permanent magnet DC motor" (PMDC) with a stator of permanent magnets with speed controlled by armature voltage and a mechanical part modeled by a load with friction and inertia (Ong, 1998).

The system model is:

$$V - V_{ce} = Ri + L\frac{di}{dt} \qquad (1)$$

where V is the armature voltage; R and L are the motor's resistance and inductance. V_{ce} is the back

(counter)-e.m.f. of the motor i.e. the voltage generated across the motor's terminals as the windings move through the motor's magnetic field. The back-e.m.f. actually opposes the drive voltage and is proportional to the motor's velocity :

$$V_{ce} = K_{cem}\frac{d\theta}{dt} \qquad (2)$$

The motor converts the electrical armature current into a mechanical torque applied to the shaft trough the torque constant C_p.

$$T = C_p i \qquad (3)$$

The applied torque produces an angular velocity $\omega = d\theta/dt$ according to the inertia J and coulomb friction B of the motor and antenna.

$$T = J\frac{d^2\theta}{dt^2} + B\frac{d\theta}{dt} \qquad (4)$$

so that the system can be modeled by the transfer function

$$\frac{\theta(s)}{V(s)} = G(s) = \frac{C_p}{s(LJs^2 + (RJ+LB)s + C_pK_{cem})} \qquad (5)$$

or the normalized parameter description:

$$G(s) = \frac{K}{s(s^2 + 2\delta\omega_n s + \omega_n^2)} \qquad (6)$$

with K the static gain; δ the damping factor and ω_n the natural frequency.

Additionally to this linear model the student can introduce nonlinear parts to the model such as dead-zone
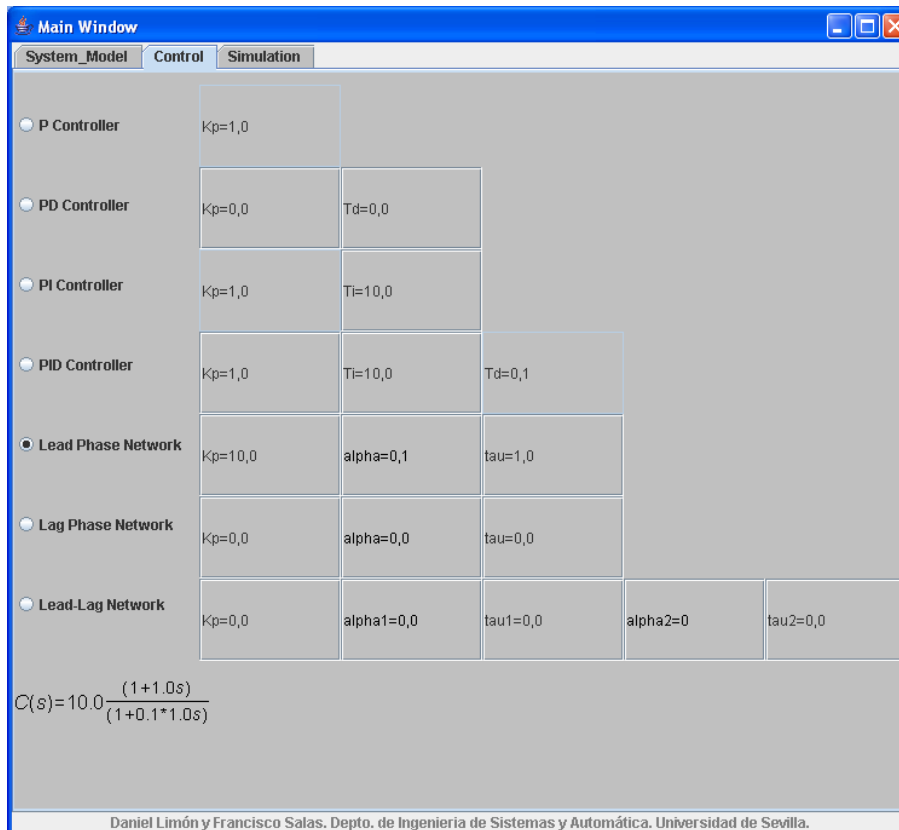
Fig. 2. Main Window: Control panel

and/or saturation in the actuator, output disturbance and dynamical nonlinearities by means of parametric uncertainties.

4. PROGRAM DESCRIPTION

The java based application simulates the open loop and the closed loop behavior of a servomechanism. This program has been developed using the Easy Java Simulation which allows one to add interaction with the user during the simulation (Limon and Salas, 2003).

The program has been designed in two windows:

Main Window : it is the Graphical User Interface of the simulator and allows the user to configure the system to be tested. This is divided in three panels: one to describe the system to be controlled (System panel), one where the controller is selected and tuned (Control panel) and the last one where the simulation is carried out in an interactive fashion (Simulation panel).

3-D view Window where the physical system, an antenna, is shown in motion in a realistic way.

In what follows, the panels of the main window are described.

4.1 System model panel

In this panel the user describe the system to be controlled by means of the defining parameters of the model (see figure 1). These parameters can be introduced in the two aforementioned forms: real and normalized parameters. The transfer function is displayed according to the chosen way of parameter introduction. An additional help has been added to the window about the transfer function of the system and their parameters.

Moreover, in this panel the student can also add the nonlinearities by marking the corresponding checkbox (ON/OFF) and configuring them by introducing their defining parameters: the value of dead-zone, the saturating value and the percentage of variation for the damping factor.

It is worth remarking that the different forms of the parameters are appropriate to introduce models obtained by identification of a real servomechanism.

4.2 Control panel

This panel is shown in figure 2 and shows the different controllers that the student can use to control the described system (PID classic controllers and lead, lag and lead-lag phase network (Ogata, 2002)).

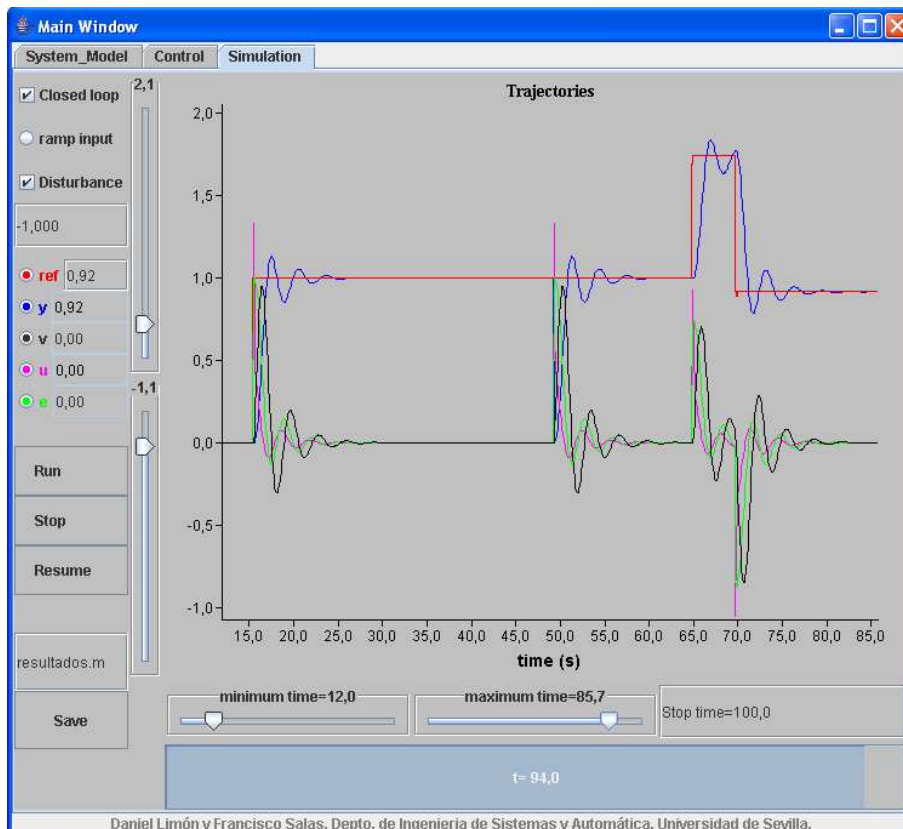The defining parameters of the controllers and networks (Ogata, 2002) can be introduced. One of these

Fig. 3. Main Window: Simulation panel

controllers can be selected by marking in the checking box close to the controller. The selected controller is implemented in the closed-loop system and the user can modify the controller parameters or change of controller while the simulation is running.

Moreover some help has been added showing in the bottom of the panel the transfer functions corresponding to the selected controller.

4.3 Simulation panel

In this panel the simulation is configured and controlled in an interactive way and the evolution of the signals of the system are depicted. Hence this is the panel where the program user can see the results of his or her control design and the effect of the nonlinearities introduced in the system model panel (see figure 3).

Given that the simulator has been designed for educational purposes, it is very important not only showing the graphics commonly used in control such as trajectories, but also a realistic representation of the evolution of the controlled system. This is achieved in the 3-D view window, linked to the simulation panel, which will be described later on.

A second requirement in the design of this panel was to provide a set of interactive elements that could be manipulated by the students to make changes in a dynamic way during an experiment.

The panel is therefore structured in three different parts. The main part is located in the right-hand side of the panel. In this part the time response is shown. There are four sliding controls to set the limits of the display and a sliding bar showing the time variation. In this part it is possible to fix or change the stop time for the experiment.

In the upper-left part of the panel user can check different choices of the simulation. These are:

- Closed-loop or open-loop simulation
- If the reference is constant (by default) or a ramp signal (of unitary slope)
- If a constant disturbance is added to the output of the signal and the amplitude of this signal.

In this part the user can also select the variables to display in the main part by selecting the appropriate checkbox (ref: Reference for the angular position of the antenna (rd); y: Main output, angular position of the antenna; v: Antenna velocity; u: Control signal; and e: position error).

The bottom-left part of the simulation panel is filled with three buttons that allow the user to manage the program. With these elements the user can select to *Run*, *Stop* or *Reset* the experiment.

Once the simulation results are satisfactory all the variables displayed can be *saved* (in plain text format ready to be read by Matlab) to be analyzed and displayed with other programs.

4.4 3-D View window

This is shown in figure 4. As it has been stated before, one requirement of the simulator is to provide a visualization that displays the physical system as realistically as possible. So that, in this window an animation of the antenna is given.

The set point can be changed interactively by dragging the sliding control left or right or with the reference value in the simulation panel. The desired position of the antenna is depicted by a red bar in the 3-D view and thus, the objective of the control is to reach this position with a suitable evolution.

This window helps the student to understand the system behavior and the effect of the different controllers and nonlinearities present in the system.

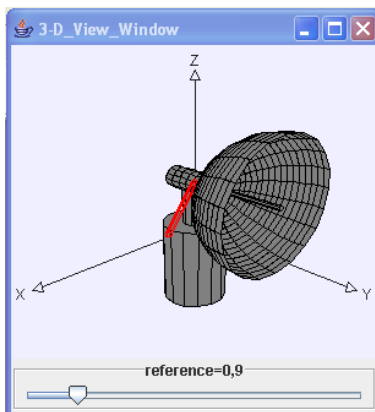User can change the point of view by dragging the antenna with the mouse pointer.

Fig. 4. 3-D View window

5. CONCLUSIONS

In this paper it has been shown an application to be used for education in control of lower courses. This application has been designed by means of Easy Java Simulation. The objective of this simulator is to help the student to understand and get matured the control problem and the usage and design of classic controllers. The main properties of the developed application that can be highlighted are its interactive nature, the multi-platform application, the availability of the application by www or downloading and installing (Limon and Salas, 2003).

The simulator allows the user to configure the system, the desired controller and simulate the closed loop evolution of the system showing a 3-D view of the antenna. Moreover usual real nonlinearities and disturbances can also be added.

This application has been successfully used in basic control courses by the authors at the University of Seville for the last three years.

REFERENCES

Dormido, S. (2004). Control learning: Present and future. *IFAC Annual Control Reviews* **28**, 115–136.

Dormido, S., C. Matrin, R. Pastor, J. Sanchez and F. Esquembre (2004). Magnetic levitation system: A virtual lab in "Easy Java Simulations". In: *American Control Conference*.

Esquembre, F. (2002). Easy Java Simulations 3.1. *http://fem.um.es/Ejs*.

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

Heck, B.S. (1999). Special report: Future directions in control education. *IEEE Control System Magazine* **19, No 5**, 35–58.

Limon, D. and F. Salas (2003). Virtual lab of servomechanism. In: *http://www.esi2.us.es/~fsalas/en/LabVirServo.html*.

Ogata, K. (2002). *Modern Control Engenieering*. Prentice Hall. Fourth Edition.

Ong, Chee-Mun (1998). *Dynamic Simulation of Electric Machinery using Matlab/Simulink*. Prentice Hall Ptr.