International Journal of Engineering Education Vol. 27, No. 3, pp. 528–534, 2011
Printed in Great Britain

0949-149X/91 $3.00+0.00

Understanding Automatic Control Concepts by Playing Games*

J. SÁNCHEZ, S. DORMIDO-CANTO, G. FARÍAS, F. GODOY and S. DORMIDO
Dept. Informática y Automática, UNED, C/. Juan del Rosal 16, 28040 Madrid, Spain. E-mail: {jsanchez, sdormido}@dia.uned.es

This paper presents three simulation-based games developed by the Department of Computer Science and Automatic Control at the Spanish University for Distance Education (UNED) to aid in the teaching/learning process of basic control concepts. The learning subject of the games is a PID controller, which is a control element that is commonly included in the syllabus of an introductory course on automatic control. The games are part of the results of a European Leonardo da Vinci pilot project entitled 'AutoTECH: Automation Technicians Vocational Training Repository'. The goal of this project is to develop and disseminate innovative software packages for technicians and vocational students in the automation and control field. These training packages will improve the quality of learning of these two groups of people and increase their motivation to acquire new knowledge related to their professional careers. Each of the packages contains both traditional (e.g., theory, exercises, quizzes) and web-based (e.g., interactive simulators, remote operable labs, games) learning resources.

Keywords: control education; learning-by-playing; dynamic simulations; Java

1. Introduction

Constructivist educational theory focuses on conceptual development and understanding as goals of instruction rather than behaviours or skills; additionally, development and understanding are constructs of active learner reorganisation. The fundamental principle of constructivism is that knowledge is not directly transmitted from one student to another but is actively built up by one's self.

Learning involves individual constructs of knowledge that result from interaction with the environment or culture. Therefore, students are considered to construct their own knowledge of the world [1].

Students should also accept responsibility for their own learning and be self-motivated to explore various domains of knowledge. Games can affect cognitive functions and motivation [2], inherently stimulate curiosity [3] by including challenges [4], and promote goal formation and competition [5].

Computer games can be helpful in education [6]. A computer game is essentially a game composed of a computer-controlled virtual universe that players may interact with to achieve a goal. Characteristics of such games include a high degree of interactivity, advanced graphics, a highly dynamic virtual universe, and an incentive system to promote prolonged and advanced use. However, a warning should be considered: students attempt to learn and solve problems by playing without understanding. For this reason, games should challenge, attract, and encourage students to make observations and relate them to the theory to develop a broad and deep understanding.

Computer games provide an opportunity to learn complex systems by using interactivity and visualisation. These concepts are critical aspects in control engineering education [7–8]. In control engineering simulations, a typical analysis of the response of a system originates from the features of its output signals (waveform, period, gain, etc.). Because the output signals are not human-readable, the analysis of the response of the system is neither direct nor intuitive. Without a suitable visualisation, simulations can be learning objects that are difficult to understand. Furthermore, many of the analyses are performed in an offline or static manner; the signals are obtained and observed only when the simulation is complete, and students rarely interact with the system or change parameters or inputs while the simulation is running. This passive role of the students significantly slows the learning process [9].

This paper presents simulation-based games developed by the Department of Computer Science and Automatic Control at UNED to assist in the teaching/learning process of basic control concepts. The learning subject of the games is a PID controller [10], which is a control element that is commonly included in the syllabus of an introductory course on automatic control. The games are part of the results of a European Leonardo da Vinci pilot project entitled 'AutoTECH: Automation Technicians Vocational Training Repository' [11]. The goal of this project is to develop and disseminate innovative software packages for technicians and vocational students in the automation and control

* Accepted 15 October 2010.

field. These training packages will improve the quality of learning of these two groups of people and increase their motivation to acquire new knowledge related to their professional careers. Each of the packages contains both traditional (e.g., theory, exercises, quizzes) and web-based (e.g., interactive simulators, remote operable labs, games) learning resources. This paper focuses only on the games, their development, and their evaluation.

The games were developed to demonstrate to students that control devices are involved in many areas of everyday life. The primary difference between these games and classic games is the mathematical models that are used in them. In modern games, there is always a moving element (e.g., a dart, a spacecraft, a skier) in which the dynamics correspond to a first- or second-order process. The movement of the agent must be controlled automatically by using a PI or PID controller. Therefore, the player is responsible for correctly tuning the controller parameters to achieve the goals and increase his/her score.

The tool that was used to create these games is Easy Java Simulations (EJS). EJS is a free software package used to create interactive simulations in Java. This is the first time that this tool has been used for developing games. The games are a part of the full AutoTeach learning repository and are located in a learning portal named PIDstop [12], which was developed at the NTNU, Norway. PIDstop is a special-purpose learning content management system (LCMS) that is designed to fulfil the requirements for automation and control purposes.

This paper is organised as follows. Section 2 introduces the basic concepts of a PID controller. Section 3 describes EJS as a tool to build learning games. Section 4 summarises the three games that were developed. Section 5 presents an educational evaluation of the use of these games by a group of automatic control students. Finally, Section 6 puts forward the main conclusions.

2. Basic concepts of a PID controller

In the past fifty years, great advances have been made in the area of automatic control, electronics, and computer science, which has facilitated the use of complex control algorithms in real applications. However, for the regulation of industrial processes, a PID controller is the most common type of controller. PID controllers are commonly referred to as the 'bread and butter' of control engineering practice. To investigate the reason that modern control algorithms have not replaced PID controllers, it is not only necessary to consider two of the primary advantages of a PID (robustness and intuitive relationships between their parameters and the response of the system), but also its great flexibility due to recent technological advances.

A basic PID controller combines three control actions: proportional (P), integral (I), and derivative (D), which is described by the following control algorithm:

$$u(t) = K_p e(t) + \frac{1}{T_i}\int e(t)dt + T_d\frac{de(t)}{dt} = P + I + D$$

where $u(t)$ is the control signal or manipulated variable at time t, $e(t) = r(t) - y(t)$ is the error signal or deviation between the reference or set point $r(t)$ and the controlled variable (process output) $y(t)$, and K_p, T_i, and T_d are parameters for the P, I, and D actions, respectively.

The primary characteristics of each basic control action are the following: the P action varies instantaneously with the error $e(t)$, and it reaches a steady state if the error is also in steady state; the I action continuously registers the error evolution, and it reaches a steady state if the error is zero; the D action predicts the changes in the error, and it is zero if the error reaches a steady-state value. The selection of the PID controller parameters involves a trade-off between the requirement for fast control and the need for stable control. Table 1 shows, among other characteristics, the change in stability and the velocity of the process as the parameters are modified. Note that Table 1 contains only heuristic rules; there might be exceptions.

3. Using Easy Java Simulations to create interactive games

In this section, the authoring tool, Easy Java Simulations, is briefly described [13–14]. Additionally, a

Table 1. Heuristic rules of the *PID* control parameters

	K_p increases	T_i decreases	T_d increases
Stability	decreases	decreases	increases
Velocity	increases	increases	increases
Steady-state error	not eliminated	eliminated	not eliminated
Error area	decreases	decreases to a certain point	decreases
Control disturbance	increases abruptly	increases gradually	increases very abruptly

simple example that describes the use of this tool to create an interactive game is shown.

3.1 Easy Java simulations

EJS organises a Java application (games or simulations) in two main components: the model and the view. EJS also adds an introductory component to these two main components, which will not be described here. The model describes the simulated system by using variables (both state variables and parameters), which completely characterise the system, and computer algorithms, which state the evolution of the system in time and its response to user interaction. The user must declare the variables using a simple table and write the Java code required to specify the algorithms. EJS offers specialised help to solve models based on ordinary differential equations by providing a built-in editor to write these equations and automatically generating the required code using the most common solvers.

The second component, the view, provides a visualisation of the simulated system (either in a realistic form or using one or several graphs) and the user interface elements required for user interaction. The view elements can be chosen from a set of predefined, ready-to-use components to build a tree-like structure in a type of block construction game for the view.

There are elements of several types; each type specialises in a particular visualisation or interaction task, but it can also be customised using *properties*, which are a set of internal values that modify the aspect and behaviour of the element on the screen. The job of the user when building the view consists of choosing the correct elements from those available and customising them to reach the desired level of interactivity (see Fig. 1).

Both the model and the view must be interconnected. A change in the model state must immediately be reflected in the view to maintain a dynamic real-time visualisation of the system. In turn, any interaction of the user with the view must immediately affect the model so that the intended level of interactivity is achieved. This communication is based on the connection of model variables to view element properties. This connection is easily established by typing the names of the model variables in the table of properties of the view elements that we want to connect to the properties.

Once the model and the view have been created and the required connections have been established, EJS creates the ready-to run simulation with a single mouse click and takes care of a number of technical issues in a process that is completely transparent to the user. The result is an independent, high performance, interactive Java application that can be either executed as a stand-alone program or embedded as an applet in an HTML page.

3.2 An illustrative example

A simple example to describe the use of EJS is presented. Suppose that we want to create a simulation of a water tank in which the liquid level is controlled by a PI controller. The user interface of the simulation is presented in Fig. 2. The visual

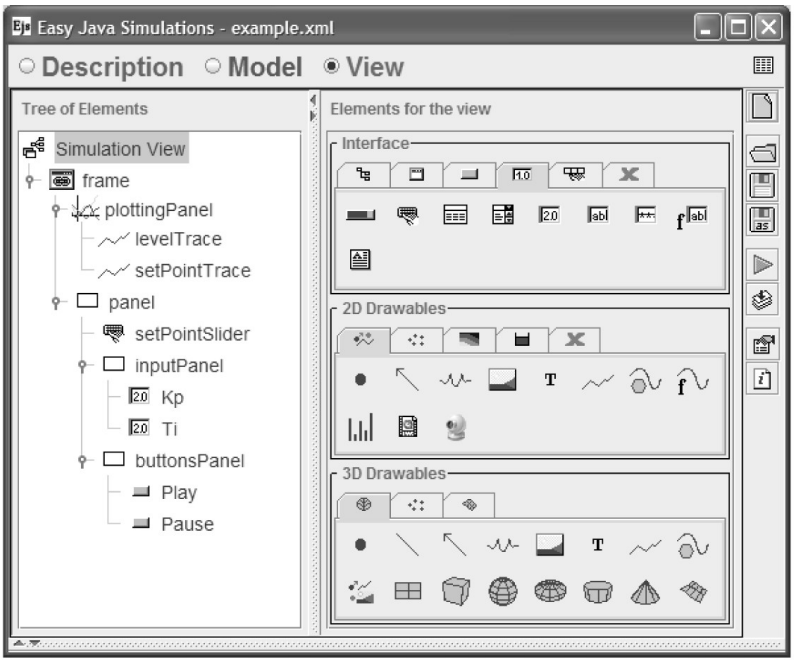


Fig. 1. The view component of Easy Java Simulations.

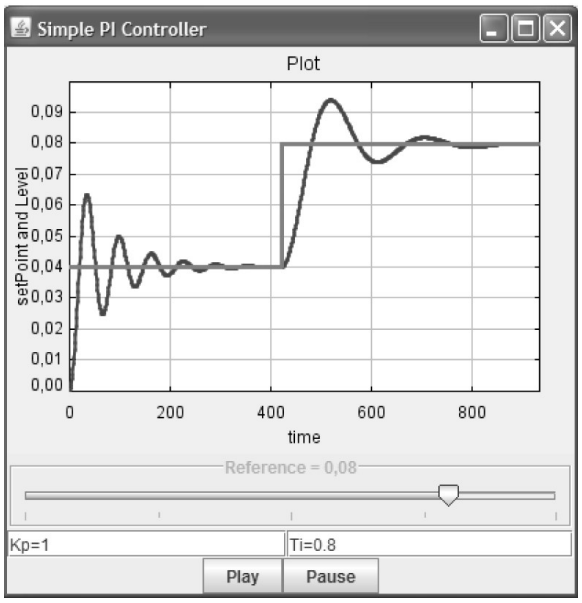


Fig. 2. User interface of the simulation of a water tank controlled by a PI controller.

elements used for creating the user interface are shown in Fig. 1.

The simulation allows students to set the control parameters in two text fields: proportional gain (K_p) and integral time (T_i). The reference (*setPoint*) can be modified with the slider. The simulation plots the reference (in red) and the level of the system (in blue). At time t = 400, the response of the system improves. This is a consequence of the modification of the proportional gain value from 8 (initial value) to 1.

The dynamics of the system and the PI control algorithm are introduced in two Evolution pages of the Model component. The water tank equations are written using the built-in ODE editor (see Fig. 3a). The control algorithm, which was implemented using simple Java code, is shown in Fig. 3b.

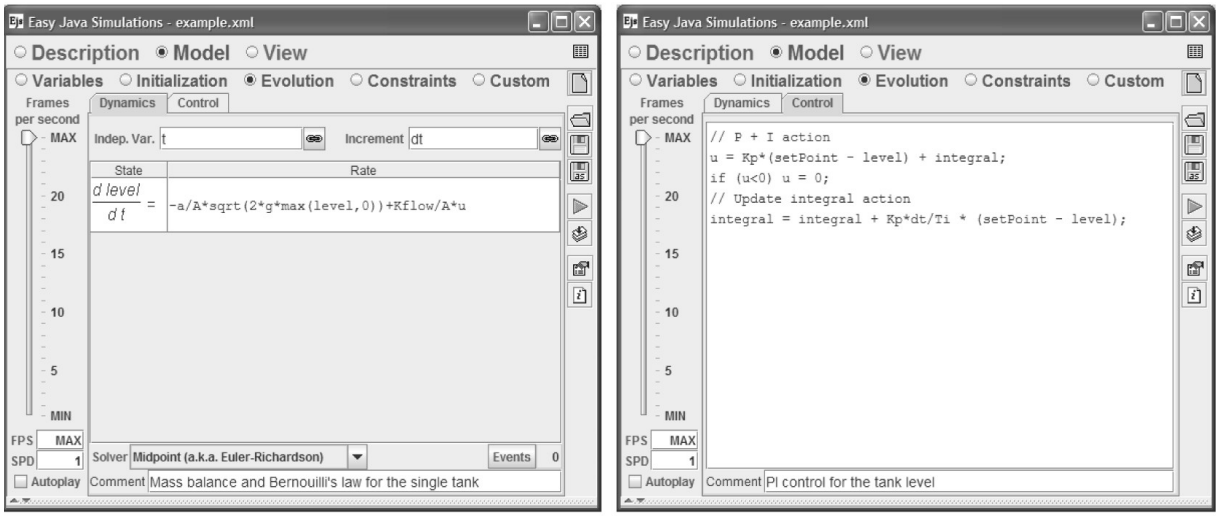
4. Developed games

By playing these games, the students can develop a broader, deeper understanding of the principles and implementation of PID controllers than they would have obtained from traditional teaching methods. After playing the games, the students will be able to investigate the effects of proportional, integral, and derivative control actions on system performance, and analyse the advantages and disadvantages of each of the control actions.

The games are projects that are given to the students to qualitatively demonstrate their ability to analyse a first- or second-order system and understand the PID control scheme. The students will be able to comment on the stability of the system (open-loop response) and determine which of the control schemes is more convenient for a given system. The three developed games are as follows: Darts, Cosmopilot, and Slalom.

4.1 Darts

The first game consists of the automatic positioning of a dart with two PI controllers. The goal of the game is to hit the dartboard as many times as possible by throwing the dart. The closer the dart is to the centre of the dartboard, the higher the score that is obtained. The dart movement is managed with two PI controllers: one for each axis. It is possible to choose between only proportional or proportional plus integral control. The PI control parameters must be modified to quickly obtain a



(a) (b)

Fig. 3. Simulation model of the water tank controlled by a PI controller.

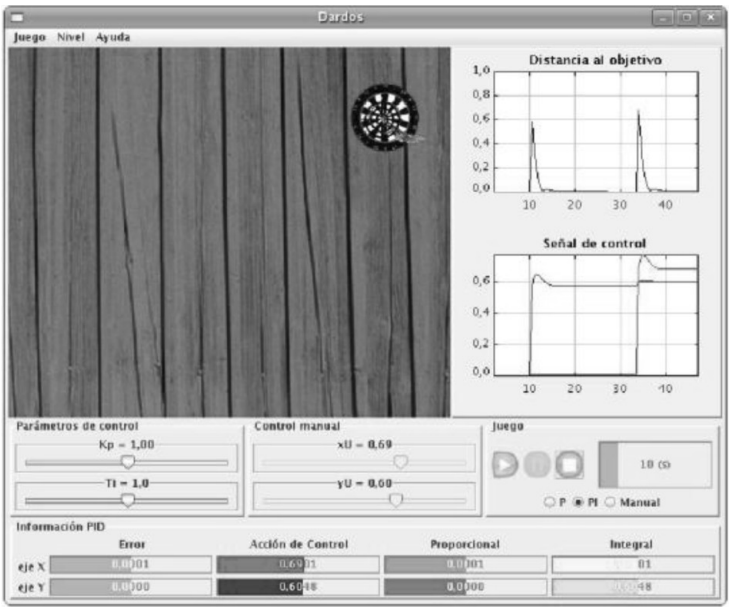


Fig. 4. Graphical user interface of the Darts game.

precise dart approximation to the centre of the dartboard. Additionally, a manual mode is available. In this case, the dart is positioned by modifying the action control of each axis.

Figure 4 shows the graphical user interface of the dart game. Sliders are located at the bottom to modify the automatic and manual controls. Additionally, information regarding the process and the PID controller parameters is provided. To the right are two graphs: one shows the reference and the distance to the goal, and the other shows the value of the control signal.

The interaction in the game requires two actions from the student. First, the student must focus on the dartboard by clicking the mouse. Then the dart is thrown by pressing any key. The movement of the dart to the selected point of focus is modelled as two first-order processes.

4.2 Cosmopilot

The second game involves the automatic control of the vertical motion of a spacecraft with a PID controller. During the match, asteroids are thrown horizontally towards the spacecraft from both sides of the screen; the user must modify the attitude reference to avoid crashing into the asteroids. Depending on the PID controller setting, which manages the thrust, the spacecraft will reach the horizontal reference either slowly or with oscillations. Additionally, there is a limited amount of time to avoid as many asteroids as possible. Fig. 5 shows the graphical user interface of the Cosmopilot game, which is similar to the darts game.

The interaction in this game allows the students to perform three actions: to control the horizontal and vertical movement of the spacecraft and to shoot asteroids to destroy them. The vertical movement of the spacecraft is modelled as a second-order process. A random disturbance (simulating an abrupt change in the gravity force) affects the vertical position of the spacecraft. The proper setting on the PID controller reduces the effects of the disturbance.

4.3 Slalom

This game allows the student to study the PID controller from a sporting point of view: this game simulates a ski race. The goal is ensure that a skier going downhill over a winding course crosses between pairs of upright poles that appear as the race advances. For this game, the dynamics of the skier correspond to a second-order process that must be controlled by a PID to avoid collisions. Fig. 6 shows the graphical user interface of the slalom game, which is similar to the previous games.

The interaction in this game allows the students to set the vertical position of the skier. It is possible to increase the difficulty level of the game by adding more obstacles. In this case, the PID controller must be tuned for a faster response.

5. Educational evaluation

The pedagogical objective of the use of these games is to observe and learn the overall effects of proportional, integral, derivative actions. A proportional controller (K_p) will reduce the rise time, but it is difficult to eliminate the steady-state error. An integral control (T_i) will eliminate the steady-state error, but it may cause the transient response to

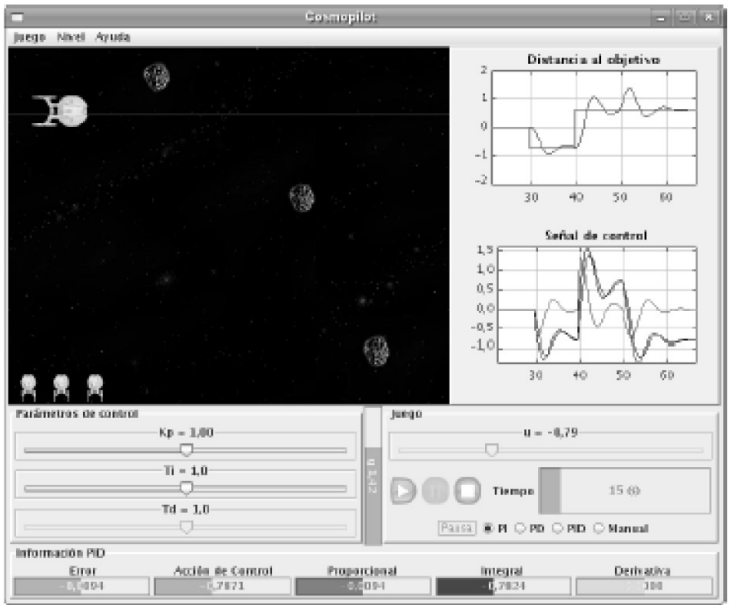


Fig. 5. Graphical user interface of the Cosmopilot game.

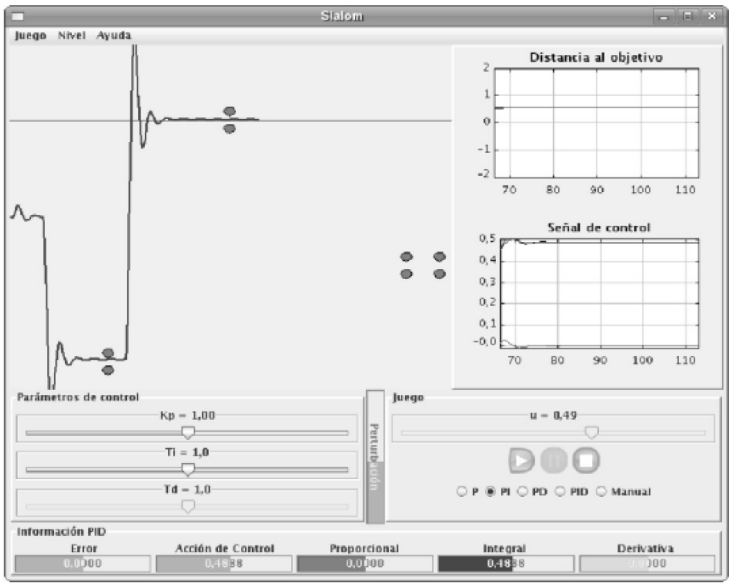


Fig. 6. Graphical user interface of the Slalom game.

worsen. A derivative control (T_d) will increase the stability of the system by reducing the overshoot and improving the transient response.

Currently, the games are available for a group of control automatic students at the AutoTECH website. The PIDstop environment retains the score of every participant student: this simple feature encourages students to improve their understanding of PID parameters to achieve better scores.

Compared to students who did not play the games, the analytical skills and performance of those who did play were superior for tuning PID control parameters. The students that played the games performed better at exams and composed well-reasoned commentaries, which improved their final marks. Fig. 7 shows a comparison between both groups: the students that played the games during the course passed the final test (i.e., marks of A, B, or C) with better results than the students that did not play the games. The results are satisfactory because the students that played the games not only obtained better marks, but they also acquired a new qualitative and practical view of their theoretical knowledge.

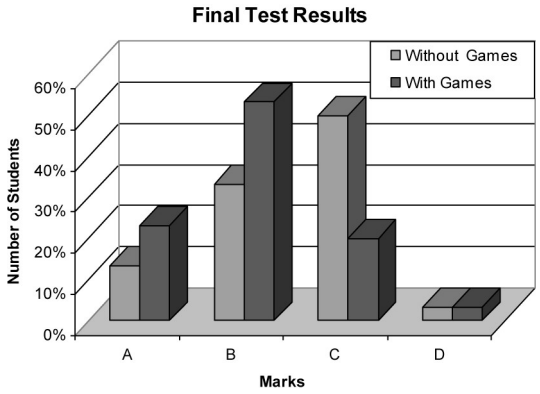


Fig. 7. Comparison of final test results.

6. Conclusions

This paper presented three educational games based on simulations that provided students with a qualitative understanding of a PID controller. The games were designed to motivate the students to review the theoretical backgrounds in a playful manner. Compared to other educational games, these games require a different didactical treatment. In particular, a scoring system was implemented to encourage students to think beforehand about how to improve the PID tunings to obtain better results. Additionally, the graphical presentations were made carefully, taking special care to maintain a consistent user interface for all of the games. As a result of these three developments, the use of Easy Java Simulations as a Java platform to develop model-based games was tested with positive results. In conclusion, EJS can definitely be accepted as a key software tool for the design and development of applications/applets for control engineering education.

Acknowledgements—This work was supported in part by the Spanish Ministry of Science and Technology under project DPI 2007-61068.

References

1. A. Amory Seagram R., Educational game models: Conceptualization and evaluation, *South African Journal of Higher Education*, **17**, 2003, pp. 206–217.
2. L. P. Rieber, Seriously considering play: Designing interactive learning environments based on the blending of microworlds, simulations and games, *Educational Technology, Research and Development*, **44**, 1996, pp. 43–58.
3. P. Thomas and R. Macredie, Games and the design of human-computer interfaces, *Educational Technology*, **31**, 1994, pp. 134–142.
4. T. W. Malone, What makes computer games fun? *Byte*, **6**, pp. 258–277.
5. L. Neal, Implications of computer games for system design in Human-computer interaction, *Proceedings of INTERACT 90, North Holland: Elsevier*, 1990, pp. 93–99.
6. M. Prensky, Digital game-based learning, McGraw-Hill, 2001.
7. B. S. Heck (ed.), Special report: Future directions in control education, *IEEE Control Syst. Mag.*, **19**(5), 1999, pp. 35–58.
8. S. Dormido, Control learning: Present and future, *IFAC Annual Reviews in Control*, **28**, 2004, pp. 115–136.
9. J. Sánchez, , S. Dormido, F. Esquembre, The learning of control concepts using interactive tools, *Computer Applications in Engineering Education*, **13**, N1, 2005, pp. 84–98.
10. K. A. Åström, T. Hägglund, Advanced PID Control, *1st edition, Research Triangle Park, NC: ISA The Instrumentation, Systems, and Automation Society*, 2005.
11. AutoTECH's Homepage:; http://www.pidstop.com/index.php?r_id=498
12. PIDstop's Homepage: http://www.pidstop.com
13. Easy Java Simulations' Homepage: http://fem.um.es/Ejs
14. F. Esquembre, Easy Java Simulations: A software tool to create scientific simulations in Java, Comp. *Phys. Comm.*, **156**, 2004, pp. 199–204.

José Sánchez Moreno is an associate professor in the Department of Computer Science and Automatic Control at UNED. His research interests include event-based control, networked control systems, and new technologies in control engineering education. Sánchez-Moreno has a PhD in the sciences from UNED.

Sebastián Dormido Canto is an associate professor in the Department of Computer Science and Automatic Control at UNED. His current research interests are related to the analysis and design of control systems via intranet or Internet, and high-performance interconnection networks for cluster of workstations.

Gonzalo Farias is a fellow student at the Department of Computer Science and Automatic Control at UNED. His current research interests include simulation and control of dynamic systems and virtual and remote laboratories. Farias is computer science engineering from Chile de la Frontera de Temuco University and has a PhD in computer science from UNED.

Francisco Godoy is a student at the Department of Computer Science and Automatic Control at UNED.

Sebastián Dormido Bencomo received his Physics degree from Madrid Complutense University (1968) and his Ph.D. from Country Vasc University (1971). In 1981, he was appointed full Professor of Control Engineering at UNED. He has supervised 35 Ph.D. dissertations and co-authored more than 250 conference and journal papers. from 2002–2005 he has been President of the Spanish Association of Automatic Control, CEA. His scientific activity includes various topics from the control engineering field: computer control of industrial processes, robust control, model-based predictive control, control education with special emphasis on remote, virtual labs and e-learning.