Int. J. Engng Ed. Vol. 24, No. 4, pp. 825–832, 2008
Printed in Great Britain.

0949-149X/91 $3.00+0.00

Design and Application of a Data Acquisition Card Simulator to Electronic Engineering Studies*

ÁNGEL PERLES*, JOSÉ ALBALADEJO, JUAN VTE. CAPELLA, JUAN MIGUEL MARTÍNEZ, HOUCINE HASSAN and CARLOS DOMÍNGUEZ
Departamento de Informática de Sistemas y Computadores. Escuela Técnica Superior de Ingeniería del Diseño. Universidad Politécnica de Valencia, Spain. E-mail: aperles@disca.upv.es

Active teaching methods in large groups of students have been applied using a simulator. This approach also allows students' work to be controlled outside the classroom, at a moderate laboratory cost. The tool is a data acquisition card simulator for the development and testing of computer applications in industrial control. It works by simulating the electrical signals connected to its pins and the physical behaviour of the processes that it controls.

Keywords: computer science for engineers; data acquisition card; simulation software

INTRODUCTION

THE INDUSTRIAL INFORMATICS TOPIC of the 2nd course of the Industrial Technical Engineers' degree in the Industrial Electronics branch of the Escuela Técnica Superior de Ingeniería del Diseño of the Universidad Politécnica de Valencia teaches students to design and develop computer-based general systems as in Fig. 1. The system interacts with industrial processes by means of sensors and actuators that are connected to an input/output interface, and it interacts with the operator by means of a human–machine interface (HMI).

We have developed a teaching approach [1] that combines accepted active learning techniques with the deliberate intention of creating a realistic working environment that simulates that which the student will experience in his or her future career. The general aims of our approach are:

- to guarantee a knowledge of the fundamentals of the subject as the necessary basis for the development of further objectives;
- to accustom the students to working as part of a team in solving industrial problems;
- to encourage the students to use practical skills in order to improve their problem solving abilities in the situations that they will meet in their working environment; and
- owing to the rapid advances in this area, to develop the capacity to adapt to any new computer-based systems that may appear in the future.

The normal teaching methods used to achieve these objectives are generally only applicable to small student groups and are expensive as regards laboratory equipment and the technicians needed to maintain it.

In our particular case, we can teach groups of around 60 students. We have successfully applied active learning techniques to these large groups and we have been able to drastically reduce the need for specially equipped laboratories and maintenance personnel. This has been partly due to the use of a data acquisition system simulator called SimSeny [2] that we have developed for this purpose. The simulator and its source code in Borland C++ Builder can be obtained at http://www.disca.upv.es/aperles/simseny.

The use of simulators as a learning tool has been applied successfully in different engineering courses [3–6]. In our case, SimSeny is software that is usable in a PC with Microsoft Windows; it replaces data acquisition cards and the physical processes connected to the cards. Nothing more is needed other than a PC and a programming environment to be able to design and evaluate applications that utilise a data acquisition card related to an industrial process.

With this simulator, any computer classroom equipped with PCs can be used for students to practise with data acquisition systems. The students are also free to practise at home, and this allows the faculty to closely direct their work outside the classroom.

The simulator gives us the considerable advantage of being able to control the work of the students outside the classroom. As part of their assessment, students have to carry out a 'mini-project' consisting of solving a set of control systems problems.

This paper is organised as follows. We begin by

* Accepted 21 January 2008.

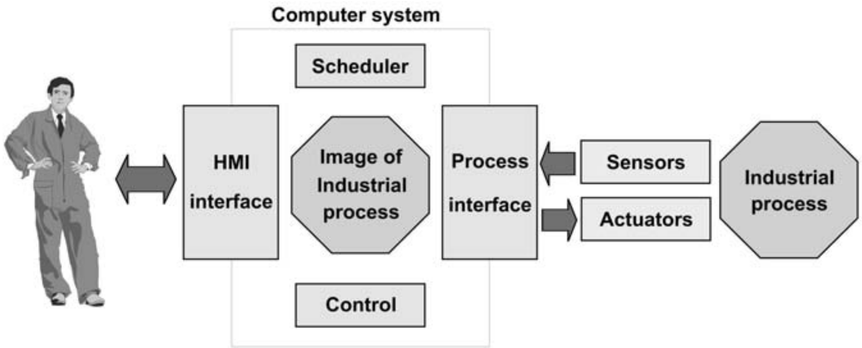


Fig. 1. Diagram of the industrial computer system.

describing the student assessment, giving special emphasis to this aspect. Next we give some details about the teaching techniques employed in this subject, which describe the 'method of the case' [7] and 'problem-based learning' [8,9] in the form of a control system case to be solved throughout the academic year. In the last part we describe the design philosophy of the simulator and also of its structure and practical applications.

THE 'MINI-PROJECT' BASED LEARNING

A good exam should make the student concentrate on the objectives of the subject. In this way, exams are a fundamental part of the teaching method and aim to impart the skills described in the Introduction.

These aims are difficult to achieve with the classic final written exams and we therefore use a combination of assessment methods. Figure 2 summarises the methods and their relative weights.

The 'mini-project' [10] is one of the most important tests for the student. It consists of a simplified project for the development of a computer-based control system, worked in a group and defended openly. This approach foments that the student acquires other abilities such as working in a group, public presentation of his ideas, technical writing, etc. [11], besides progressing in his or her speciality. The SimSeny simulator plays a vital role in the development of the 'mini-project'.

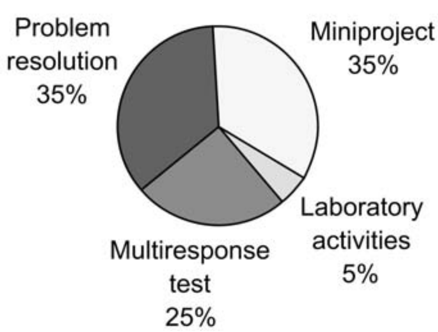


Fig. 2. Weight of each exam.

For the development of the 'mini-project', the student uses a personal computer, the Borland C++Builder programming environment for Windows and the National Instruments USB-6008 data acquisition card.

Work on the 'mini-project' is synchronised with theoretical instruction. The students deliver a series of reports on their progress and the teacher supervises the work of the group and gives feedback to help prevent possible mistakes.

The reports provided by the students are as follows.

- A paper giving the requirements of the project.
- Project specifications.
- Suggestions for dividing the project into modules.
- A description of the individual modules.
- Integration of the modules in the final application.
- A user's manual for the finished application.

These 'mini-projects' can be carried out because the SimSeny simulator allows the student to work on his or her project in a normal computer classroom or at home.

TEACHING AIMS

The designed simulator is a fundamental part of our teaching method. In this section we describe some of the key points.

In order to get the best results from our students, our methods have to take into account their limitations in working towards the desired results. The limit of the students' previous knowledge necessarily restricts the techniques and tools that can be applied, since the objectives of the subject could be mistaken for the use of certain tools or techniques. In our case, the students have a general knowledge of analog and digital electronic design and a limited knowledge of the ANSI C programming language.

To successfully deal with these limitations and to get the best from the students, they study the case of a simplified industrial control system, which is explained to them at the beginning of the course. We apply a 'problem based learning' method for

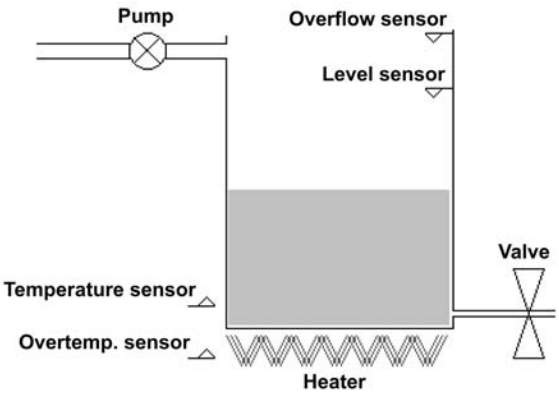


Fig. 3. Diagram of the process used for teaching.

teaching; the solutions to these problems are used throughout the course to impart theoretical knowledge and to teach the required skills.

Figure 3 is a diagram of the chosen system, which is a temperature and level control in a tank of liquid. The computer-based system must maintain the temperature of the liquid by means of a heater. Liquid can be supplied to the tank by a pump, and the heated liquid is available through an electro-valve. Level and temperature sensors supply information on the quantity and temperature of the liquid in the tank. The system is sufficiently simple for the students never to lose sight of the real aim of the classes and at the same time it teaches all the required skills.

To solve this problem we apply a 'top-down' design methodology and we use informal heuristic rules, such as 'modular decomposition' and 'hide information'. These rules allow us to propose a basic minimum structure for the application, giving rise to a minimum set of C modules that can be applied to any simple computer-based control system. Figure 4 shows this decomposition.

Each module has a specific purpose in the application, as follows:

- The *acquisition module* acts as the interface of the process. It controls the sensors and actuators by means of a data acquisition system. The physical data read on each sensor must be processed according to the characteristics of the sensor and the data acquisition system employed, and finally stored in the data module. The actuators are manipulated in the same way to reflect the desired condition represented in the data module.
- The *operator module* is the man–machine interface, and gives a graphic representation of the state of the process and allows orders to be given.
- The *control module* carries out a set of tasks in the form of C functions in a determined chronological order, thus implementing the dynamic behaviour of the application. These C functions manage each aspect of the control application, which are: the reading of sensors, the calculation of control actions, modification of actuators, and the updating of the user interface. The control module also has a function that implements a control strategy (PID regulator, fuzzy control, etc.).
- Finally, the *data module* gives a continuous image of the status of the process, e.g. the actual temperature of the liquid, the desired temperature, state of the valve, etc., and possesses a group of functions that lets modifications and queries to this image. The main purpose of this module is to avoid interdependence between modules, by centralising the shared information required by two or more modules. One of the chief benefits for this methodology is that it allows one to work on any single module without taking the others into

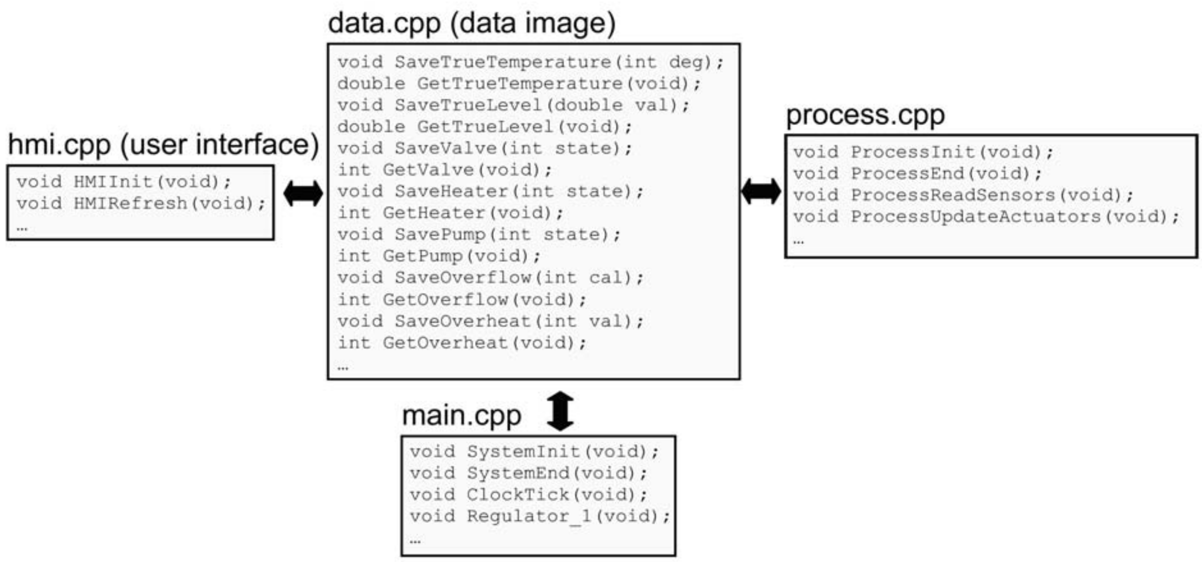


Fig. 4. Proposal for modular decomposition of the application.

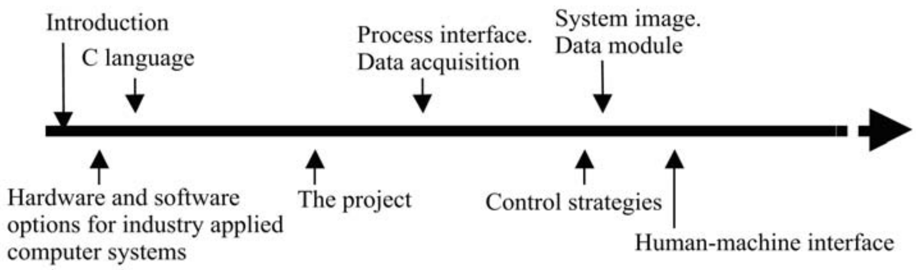


Fig. 5. Planning the subject content.

consideration. This means it is possible to concentrate on one single aspect of the problem and forget the rest.

The rules used to define these modules are those generally applied in programming. We also take advantage of the benefits of this decomposition in the teaching process by structuring the subject into easily programmed independent parts. The decomposition is useful for directing our teaching effort. Figure 5 shows this in graphic form.

Throughout the learning process, the students design and develop the tank control application. Figure 6(a) gives a possible composition of the application.

The students use the simulator to design and test the application. Having done this, they can test its operation on a scale model as shown Fig. 6(b). It is important for the students' motivation for them to have a working model available to see how their applications work on real components such as motors and valves.

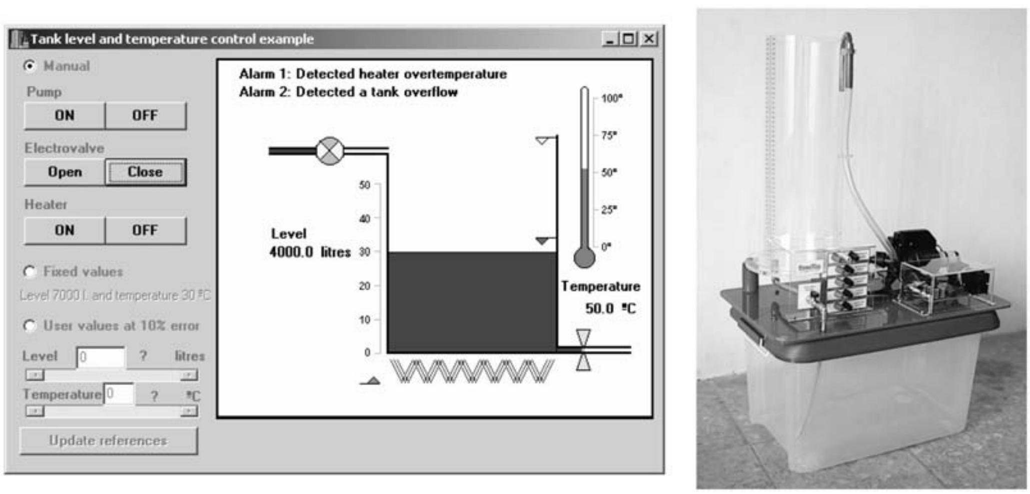
THE SIMSENY DATA ACQUISITION CARD SIMULATOR

Years of experience have shown that the greatest obstacle to obtaining effective teaching results lies in the unavailability of practice laboratories suitably equipped with computers, data acquisition cards and related processes.

We reached a point where we asked ourselves if it would be possible to develop an application for a data acquisition card without actually using a card, bearing in mind the fact that students have free access to both PCs and programming environments. We concluded that the solution would be to create a software simulator to substitute for such a card.

The simulator idea was carefully designed to achieve the following objectives:

- It had to be transparent to the student and easy to use without the need for special training.
- It had to be independent of any particular model of data acquisition card or programming language, allowing its application to different data acquisition systems and different programming environments in Windows and/or Linux operating system.
- It should permit the real or simulated execution of data acquisition in the actual or simulated performance of the application, without the need for changes.
- Most of the application parts should be able to be performed in the normal workplace, either at home or in a practice laboratory. In this way, a



(a) (b)

Fig. 6. View of application (a) and test model (b).

student's personal work could easily be supervised, since he or she could practise the fundamental aspects of the subject without needing to go to a specialised laboratory.

- It should focus the students' attention on the design and development of the application, without the need to dedicate time and effort to the assembly and testing of electronic circuits connected to a real data acquisition card.
- It should reduce the number of data acquisition cards needed, and allow the teaching method to be used with large groups of students.
- It should reduce wear and tear due to malpractice on data acquisition cards, as they are only necessary for supervised practice and final tests.
- It should reduce laboratory maintenance costs as regards personnel and materials.

The end result is a simulator that imitates the behaviour of a data acquisition card and the electrical signals that are usually connected to its external terminals. Then it is possible to validate the applications developed without any need for the card itself or for the electronic circuits connected to it.

Figure 7 shows the components of a computer-based industrial control system that uses a data acquisition card. In general, the software application interacts with the data acquisition card through drivers provided by the card manufacturer. In the case of Microsoft Windows, these drivers are usually dynamic link libraries (DLL) or ActiveX library (OCX). The card is directly controlled by the drivers, which have access to its registers. Finally, the card is connected to the industrial process through electrical connections by appropriate sensors and actuators.

Figure 8 shows a diagram of the blocks that form the simulator and how they are inserted into the industrial control system. The main idea is to capture all the calls to the functions of the card library and decide if they are to be diverted to the simulation system or, again, to the data acquisition card.

The basic components of the simulator are:

- A specific software module for the particular programming environment and data acquisition card employed.
- A dynamic link library (DLL).
- An electrical signal viewer.
- A set of physical process models.

These components will be described in the subsections below.

Adapter software module

A specific software module is required for each combination of programming language and data acquisition card. Its work is to capture all calls to the card functions made from the control application.

At the start of the control application, the user can decide if the captured cards functions still employ the real card or if they must be handled by the simulation system.

If the user decides to use the simulation system, the task of the module will be to convert the specific calls to the card to an independent format (for example, converting to volts the integer values provided to a function for handling analog output). This card-independent information is transferred to the simulation system using a dynamic link library provided with the simulator.

In our particular case, we used the development environment Borland C++ Builder and the data acquisition card National Instruments USB-6008. In order to use the simulator, the specific header file simsenyNIUSB6008.h was designed.

If we want to use the simulator in our control application, only one line has to be added to the source code C to include the above header file. For instance, the following fragment of code shows the

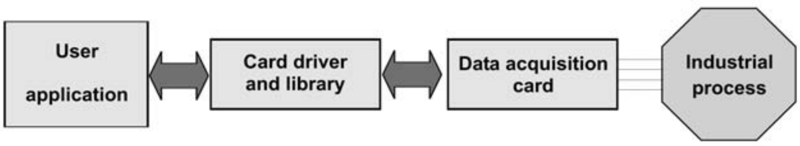


Fig. 7. Relationship between the different blocks that compose the application.

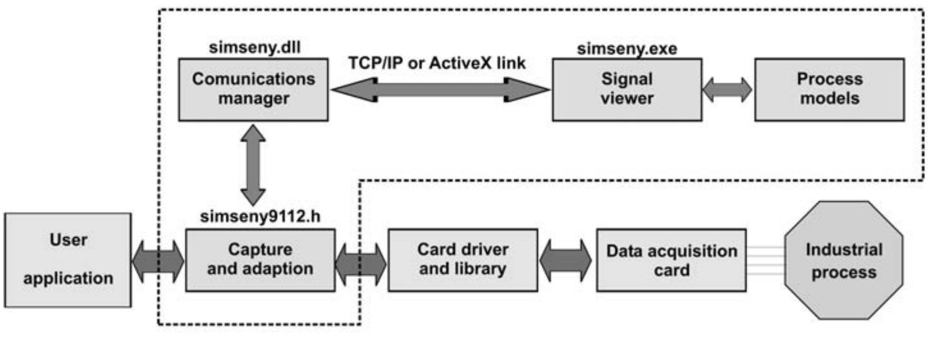


Fig. 8. Relationship between control system and components of the simulator.

functions that activate the data acquisition card and calculate the temperature of a process from the information picked up by an LM 335 sensor connected to an analog input of the card.

```
#include <NIdaqmx.h>
#include 'simsenyNIUSB6008.h'

static TaskHandle TEMP_TASK;
#define 'Dev1/ai0' TEMP_CHAN

// process module initialization
  ****************************
void process_init (void) {
  int32 error;

  // analog input for sensor temperature
  error = DAQmxCreateTask ('Temperature
   Sensor, &TEMP_TASK);
  if (error != 0) { exit(1); }
  error = DAQmxCreateAIVoltageChan
   (TEMP_TASK, TEMP_CHAN,'',
  DAQmx_Val_Diff , 0, 20,DAQmx_Val_
   Volts,NULL);
  if (error != 0) { exit(1); }
}

// read temperature sensor
  **********************************
double process_temperature(void) {
  int32 error;
  uint32 data;
  int reads_done;
  double volts, degrees;

  error = DAQmxReadBinaryU32(TEMP_TASK,
   1, 0, DAQmx_Val_GroupByChannel,
  &data,&reads_done,NULL);
  volts = data*(10.0/4096.0)- 5.0; // to
   volts
  degrees = (volts + 5.0)*10.0; // to
   degrees
  return(degrees);
}
```

Dynamic link library simseny.dll

The simseny.dll module is responsible for putting the developed application in contact with the viewing module of the simulator.

Basically, it tries to make the design of the adapter module described above as simple as possible, facilitating its development. It also tries to separate the simulator and viewing module to allow the execution of the user application in a different computer, thus enabling the application developed by the student to be remotely controlled.

This module can connect with the viewing module in two ways: either by using the ActiveX interface of the viewer or, if the viewer is required to be used remotely, by means of a TCP/IP connection.

The use of a DLL for the simulator design enables this tool to be applied to many programming environments such as Microsoft Visual C++, Borland C++ Builder, Microsoft Visual Basic, National Instruments LabView/LabWindows, etc.

Signal viewer

Figure 9 shows the *SimSeny* simulator signals viewer in operation. The viewer permits physical outputs to be shown and physical values to be entered in the simulated data acquisition card inputs, both for digital and analog signals.

This simulator interface was carefully designed to facilitate its operation. The simulated signals are:

- Digital outputs: There are 16 digital outputs, which are graphically represented by coloured circles. The circles change in colour according to the value they represent, or logic level in the output. In green, the logic value is '0', and in red the value is '1'.
- Analog outputs: There are 2 analog outputs. In SimSeny, the power of the analog outputs can

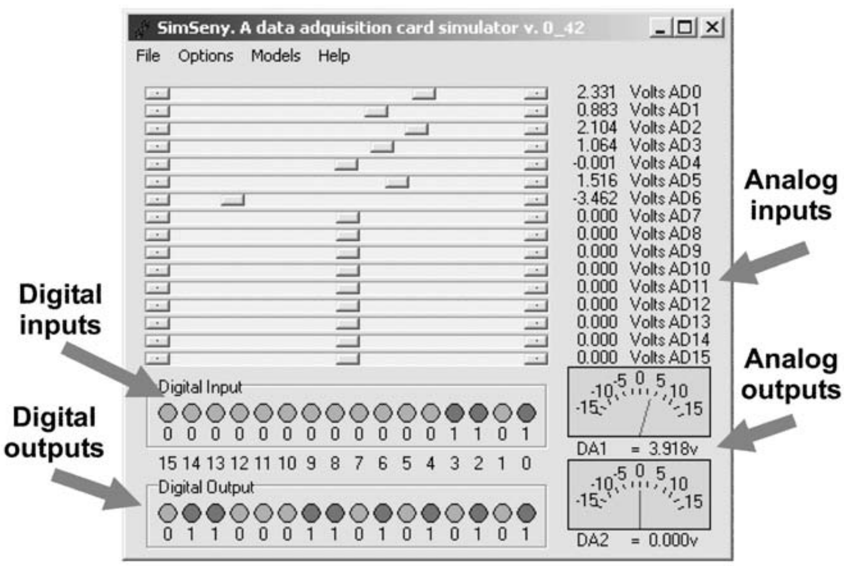


Fig. 9. Input and output signals viewer.

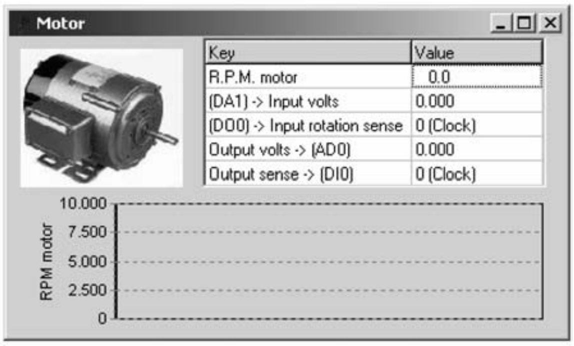


Fig. 10. Window of the process model of a DC motor speed control.

vary between [–15, 15] volts., which are represented graphically by a needle positioned on a graduated scale.

- Digital inputs: There are a total of 16 digital inputs. These are represented graphically by coloured circles. When a mouse clicks on the circles, the colours and the represented logic value changes.
- Analog inputs: There are 16 analog inputs. Any input is selected graphically in SimSeny by means of a scrollbar. To the right of each bar the analog value entered is shown.

Models of processes

As part of the teaching programme, process models that simulate the physical and electronic behaviour of certain processes are incorporated in the simulator.

When one of these models is activated on the interface of the simulator, the physical connection of the sensors and actuators of the process are imitated at the card inputs and outputs. Using the process modules, the students can create a control application and test its efficacy on the simulator.

For example, Fig. 10 shows the window of a model of a DC motor speed control connected to an encoder. The application created by the student must control the power supplied to the motor in order to reach a certain speed measured through the analog encoder.

CONCLUSIONS

The SimSeny simulator allows us to teach large groups of students without the need for costly laboratory equipment.

The system could also be easily applied to other subjects that involve the use of data acquisition cards, since the design of the simulator enables it to be adapted to other cards and programming environments.

With the simulator, the teaching methods that can be utilized allow the student to take the lead in the learning process and the teacher becomes a guide in the process.

We update the simulator for every academic course to provide a new physical process model simulation to allow evaluation of the students with the 'mini-project'.

Currently we are developing extensions to the simulator viewer, to make use of remote data acquisitions cards. This will allow a user application prepared for the simulator 'to see' and use a data acquisition card installed on a remote computer. This being possible, the student will be able to use any model of card in a transparent and remote way.

Acknowledgement—We would like to thank the Foreign Language Co-ordination Office at the Polytechnic University of Valencia for their help in translating this paper.

REFERENCES

1. H. Hassan, J. M. Martínez, C. Domínguez, A. Perles and J. Albaladejo, Innovative methodology to improve the quality of electronic engineering, *IEEE Transactions on Education,* **47**(4), 2004, pp. 446-452.
2. A. Perles, J. M. Martínez, C. Dominguez and J. Albaladejo, SimSeny: Un simulador didáctico para tarjetas de adquisición de datos, *Actas IV Congreso de Tecnologías Aplicadas a la Enseñanza de la Electrónica (TAEE2000)*, **2**, 2000, pp. 493–496.
3. M. Pinotti and D. Brandao, A flexible fieldbus simulation platform for distributed control systems laboratory courses, *Int. J. Eng. Educ.*, **21**(6), 2005, pp. 1050–1058.
4. C. S. Peek, O. D. Crisalle, S. Depraz and D. Gillet, The virtual control laboratory paradigm: architectural design requirements and realization through a dc-motor example, *Int. J. Eng. Educ.*, **21**(6), 2005, pp. 1134–1147.
5. J. Liu and R.G. Landers, Integrated modular machine tool simulation for education in manufacturing automation, *Int. J. Eng. Educ.*, **20**(4), 2004, pp. 594–611.
6. J. Liu and R.G. Landers, Modular control laboratory system with integrated simulation, animation, emulation, and experimental components, *Int. J. Eng. Educ.*, **21**(6), 2005, pp. 1005–1016.
7. P. K. Raju and C. S. Sankar, Teaching real-word issues through case studies, *Journal of Engineering Education*, **88**(4), 1999, pp. 501–508.
8. Problem-Based Learning Initiative. *Generic Problem-Based Learning Essentials* http://www.pbli.org/pbl/generic_pbl.htm
9. L. B. Larsen, S. K. Andersen, F. Fink and E. Granum, Teaching HCI to engineering students using problem based learning, *Interact Workshop of IFIP WG 13.1,* September 2003.

10. A. Perles, C. Dominguez, J. M. Martínez and H. Hassan, Enseñanza de la informática industrial mediante proyectos reales simplificados, *Actas V Jornadas Sobre La Enseñanza Universitaria En Informática (JENUI '99)*. 1.999, pp. 55–60.
11. A. W. Fentimann and J. T. Demel, Teaching students to document a design project and present the results, *Journal on Engineering Education*, **84**(4), 1995, pp. 329–333.

Angel Perles received his MS and Ph.D. degrees in computer engineering from the Polytechnic University of Valencia (UPV), Valencia, Spain, in 1994 and 2003, respectively. He has been an Assistant Professor of the Department of Computer Engineering (DISCA) at the Polytechnic University of Valencia, Valencia, Spain, since 1997. His research interests include discrete-event based modelling and simulation, real time systems, and industry-applied computers.He is a Member of the Fault Tolerant Systems Research Group.

José Albaladejo received his MS in fundamental physics from the University of Valencia (UV), and his PhD in computer engineering from the Polytechnic University of Valencia (UPV), Valencia, Spain, in 1984 and 2003 respectively. He has been an Assistant Professor of the Department of Computer Engineering (DISCA) at the Polytechnic University of Valencia, Valencia, Spain, since 1989. His research interests are fault injection, hardware and software co-design, and reconfigurability. He joined the Fault Tolerant Systems group in 2000.

Juan V. Capella obtained his degree in Computer Science in the Polytechnic University of Valencia, Spain, in 2000. He joined the Fault Tolerant Systems research Group in the Polytechnic University of Valencia, where he received the MS degree in Computer Engineering, in 2002. He has been an Assistant Professor of the Department of Computer Engineering at the Polytechnic University of Valencia since 2002. His main research interests are fault tolerant systems, distributed systems design and wireless sensor networks. He has been involved in different innovation projects in higher education since 2002.

Juan-Miguel Martínez received his BEng degree in electrical engineering and his MS and Ph.D. degrees in computer engineering from the Polytechnic University of Valencia (UPV), Valencia, Spain, in 1986, 1993, and 1999, respectively.He has been with the Department of Computer Engineering, UPV, since 1986, where he is currently an Associate Professor. His research interests include power saving, deadlock handling and congestion control mechanisms for computer interconnection networks. He has been involved in different projects in innovation in higher education since 1990.

Houcine Hassan received his MS and Ph.D. degrees in computer engineering from the Polytechnic University of Valencia (UPV), Valencia, Spain, in 1993 and 2001, respectively. He has been an Associate Professor of the Department of Computer Engineering at UPV since 1994. His research interests focus on a number of aspects of the development of hardware and software architectures for embedded systems, real-time systems and robotic systems. He joined the Industrial Informatics group of the UPV in 1993, where he is presently participating in several research and educational projects.

Carlos Domínguez received his MS degree in electrical engineering from the Polytechnic University of Valencia (UPV), Valencia, Spain, in 1991. He has been an Assistant Professor of the Department of Computer Engineering, UPV since 1992. His research interests include the development of intelligent agents for mobile robots, real-time systems, and planning and scheduling integration. He has been involved in different projects in the Informática Industrial group since 1989.