

Development of a Wiimote-based gesture recognizer in a microprocessor laboratory course

Alberto Lorente-Leal
Dept. Ingeniería Electrónica
Universidad Politécnica de Madrid
Madrid, Spain
alorleal@ieee.org

José Á. Fernández-Rodrigues
Dept. Ingeniería Electrónica
Universidad Politécnica de Madrid
Madrid, Spain
joseangel@ieee.org

Juan M. Montero
Dept. Ingeniería Electrónica
Universidad Politécnica de Madrid
Madrid, Spain
juancho@die.upm.es

Abstract— This gesture recognizer, developed by students in a third-year microprocessor-based laboratory course, takes Wii remote (Wiimote) as an input device to estimate the movements of the user and to compare the detected trajectory with the previously learnt movements, in order to carry out the associated actions. Such a cheap state-of-the-art wireless user interface is very attractive for the students and can be used in many interactive applications, from robotics to virtual reality and multimedia presentations. By combining commercially-available hardware, pattern-matching techniques and programming skills, we are able to foster students' interest on developing innovative potentially-marketable systems. This freeware project, implemented as a configurable publicly-available library, can be adapted to the needs of any course or student. In our laboratory this open-source DLL is used for remotely controlling a robot (based on an open-hardware Arduino platform), using a PC and the Wiimote, although the DLL can be integrated in any C, C++, Java or C# project. A GUI application (based on a Model-View-Presenter paradigm) is also provided and can be used as a template for new applications or just for debugging purposes. Although the developed application only uses data from the accelerometers, data from the infrared camera and buttons of the Wiimote is also available.

*Microprocessor; Wiimote; Project Based Learning; open source library; microprocessor programming; Arduino*¹

I. INTRODUCTION

Project Based Learning (PBL) is a systematic teaching method that engages students in learning both theoretical knowledge and practical skills through an extended inquiry process structured around complex, authentic questions and carefully designed products and tasks [1]. The 21st century information and knowledge economy is shaping the way in which colleges and universities are preparing undergraduate and graduate students with knowledge skills and abilities throughout the world. PBL is becoming one of the most important educational approaches to help faculty improve student learning process and there are many examples of its successful application in university courses [2] [3]. In

This work was supported in part by UPM under innovation projects

university teaching it has been applied to a wide variety of disciplines from business to science [4] [5]; but most applications have been in engineering courses [6] [7]. Compared to traditional teaching, PBL technique reveals a higher degree of learning [8] [9]. PBL allows increasing students' involvement in the learning process, obtaining better results in terms of the theoretical knowledge and practical skills students are able to acquire.

In our PBL laboratory course LSED, students must face a State-of-the-art project based on a commercial Wiimote interface and R&D and innovation pattern-classification techniques.

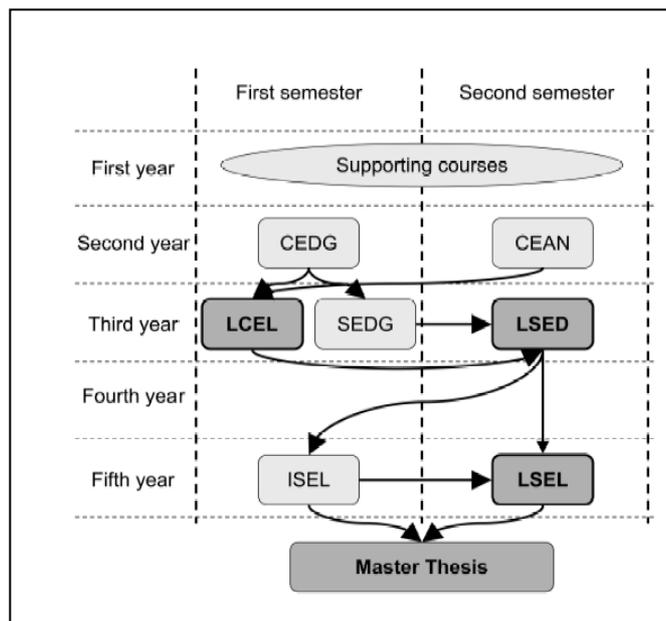


Figure 1. Structure of the curriculum on Electronics.

The paper is organized as follows: Section II describes the laboratory course in the context of the curriculum structure. The implementation of the system is described in the following sections. First the required pattern-recognition algorithms are explained; afterwards hardware and software requirements are outlined. Finally the conclusions of the proposal are drawn.

II. COURSE DESCRIPTION: LSED

In the literature, many curriculum initiatives used PBL as their main learning strategy [4]. In some cases, PBL is enhanced using multiple case-studies or using several mini-projects for improving assessment [5]. In LSED case, students must develop a multidisciplinary supervised project.

The curriculum scheme can be seen on Figure 1. CEDG, CEAN, SEDG and ISEL are theoretical courses while LCEL, LSED and LSEL are project-based ones. The arrows show the dependencies between subjects from a theoretical point of view.

A. Theoretical courses

The three theoretical courses before LSED are problem-oriented and examination-based, with a great emphasis on analysis issues, albeit the evaluation is mainly based on the analysis of a simplified system.

CEDG (Digital Electronic Circuits) and CEAN (Analog Electronic Circuits) introduce the fundamentals of both digital and analog circuits, both in the second year (3rd and 4th semester). The gesture recognizer described in this paper can control a hardware system that can be analyzed by means of the fundamentals provided in these courses. An example (a robotic system) is described in section VI.

In SEDG (Digital Systems Based on a Microcontroller): students learn the fundamentals for programming a microcontroller-based system: inner structure, programming issues, timing resources, exception handling and peripherals (such as universal asynchronous receiver/transmitter or analog-to-digital converters). The emphasis is put on system-level programming.

After LSED, one theoretical course is left (ISEL, Electronics Systems Engineering), focused on advanced embedded systems, operating systems and real-time issues.

B. Project-Based courses

The design of electronic systems must deal with decisions in several axes:

- HW versus SW design (or co-design).
- Life cycle process: analysis, design, implementation, and testing iterations.

To involve student in dealing with all these aspects more than one semester is needed. PBL courses must be coordinated to get complementary learning objectives covering the whole life cycle and all hardware/software aspects.

In these courses, the teaching approach is based on multidisciplinary PBL, using electronics to create systems to solve problems in signal processing, communication or control areas. Students must design, build, test, and document a complete system (hardware or software or both of them) with emphasis on creativity (innovative solutions), realism (in terms of cost) and professionalism (evaluating factors such as technical writing quality and oral communication capabilities).

LCEL (Electronics Circuits Laboratory) is the PBL course that precedes LSED, focused on the design and implementation of analog and digital electronic circuits from a set of technical specifications.

LSED (Microprocessor-based Laboratory), in the first semester of the third year, students must develop an electronic system mostly based on programmable devices. The system always includes a real-time component (in the proposed Wiimote system, data from the device must be obtained and processed in an interrupt service routine), making the debugging of the system more complex and thus the development of the whole prototype more challenging.

After LSED, only one optional PBL laboratory remains: LSEL (Laboratory of Electronics Systems Engineering), focused on a software/hardware balance, reusing previously-implemented modules when implementing a fully-functional prototype. Programming is not a critical issue and the emphasis is on high-level programming and student-defined specifications.

C. Master's Thesis.

Finally, every student must complete a Master's thesis individually in order to get the Telecommunication Engineering degree. At this point, students coming from LSEL are very well prepared to take on a Master's thesis involved in one of the research and development lines developed in the Department of Electronic Engineering.

Similar to LSEL, the Master's thesis focuses on both SW and HW aspects and considers all the steps in the life cycle of the development of an electronic system.

The main difference with respect to LSEL is that Master's theses are developed individually: The student has to analyze, design, and implement the system architecture, controlling the functionality of all the modules. First of all, the student has to design a detailed testing plan, which is one of the main targets in this level of the curriculum. As a result of the Master's thesis, the student develops a quasi-professional system, generally involved in a research and development project that is being completed by the instructors with a company or a government institution.

III. WIIMOTE DATA SAMPLING AND END-POINTING

A sample of data received from a Wiimote is shown in Figure 6. Three different regions are shown in Figure 3.

Region 1 corresponds to a situation when no intended Wiimote movement is being made, just random movement or electronic noise is received. In this region 1, the received values vary between zero and the first threshold (green line). If an intended movement is carried out, the data values pass through an intermediate region 2, a transition area between "no detected movement" and "a movement clearly detected" (labeled in the diagram with a 2). The system detects the beginning of the movement when the data values overpass a second threshold for a brief period of time. Then, collected data is identified as a specific gesture (region 3) until data return to a noisy situation which is reached after a contiguous sequence

of samples are detected below the first threshold (final region 1).

This end-pointing algorithm is implemented as a software finite state machine as shown in Figure 2.

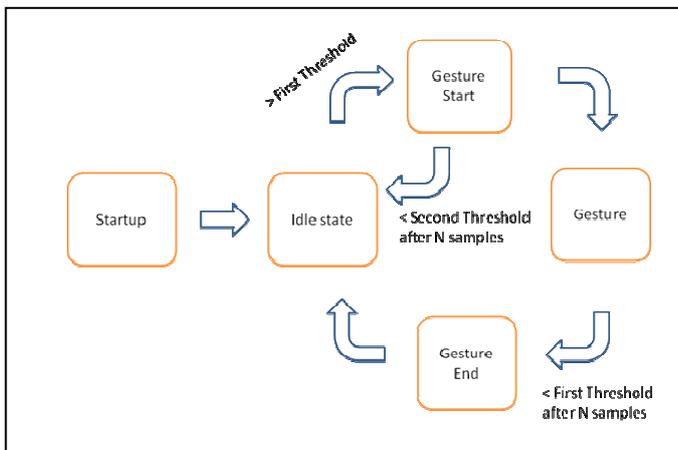


Figure 2. Finite state automaton of the gesture recognition system

The first state is Startup, which calibrates the system in order to take into account electronic noise and working conditions when establishing the values of the thresholds. This way, the system can automatically adapt to the user that is using our recognition software at that specific moment in a specific environment.

After this calibration, the Idle state starts; Wiimote data is sampled for later comparison and to know if any activity has occurred above our first threshold. If first threshold is exceeded and the system does not exceed the second one after several samples, a false alarm has occurred, no gesture should be saved and the system must go to the previous state again.

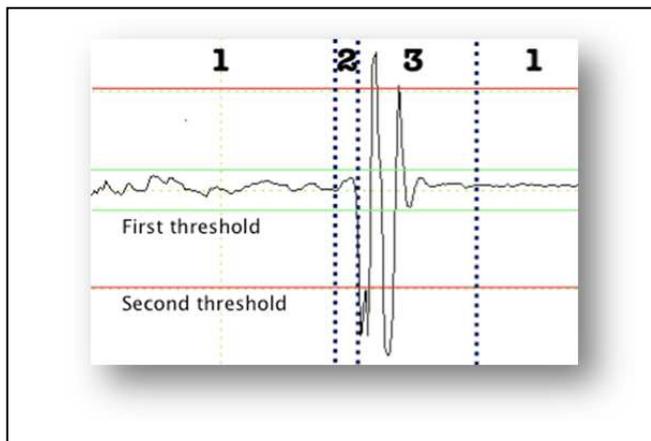


Figure 3. One-axis acceleration data when a movement is performed

If the second threshold is surpassed, a new gesture is actually being made, so the system starts to save all the data received from the Wiimote unit, until data below the first threshold is received. This involves that a new “no movement

or noisy situation” has started and the movement has already finished.

IV. GESTURE RECOGNIZER SYSTEM DESCRIPTION

In order to identify the gestures we need to have previously-collected data to compare with. That is way the recognizer works, in two modes: training mode and recognition mode.

During the training phase, end-pointed movement data is serialized and saved as an XML file. This file is used in the recognition phase to compare the new data received from the Wiimote with every movement previously saved by means of the Dynamic Time Wrapping method (DTW), very popular in the design and implementation of speech recognizers for low-end mobile phones.

This DTW algorithm is able to compare two sequences of multidimensional data (such as the three-axis acceleration data from the Wiimote) to each other, even if the length of the sequences is not the same. In order to carry out the comparison, the standard DTW algorithm allows insertions and deletions in one of the sequences (although penalizing these operations when computing the distance between the sequences) as shown in Figure 5. Although normalizing the data of the sequences (such as subtracting the average value of the sequence to each of the samples) is a usual preprocessing technique (before comparison), good gesture recognition rates have been achieved without any normalization and using a linear scale and an Euclidean distance (for other signals such as speech, it is usual to use the log-energy of the signal when computing the distance, although for acceleration data, a linear approach is completely appropriate).

V. DLL AND GUI DEVELOPMENT

The Wiimote-based gesture recognizer DLL was designed using C# Programming language on the .Net 2.0 Framework [10] [11], using a Visual Studio Express 2008 (free Student Edition) to develop the architecture, as this Integrated Development Environment is quite intuitive and easy to start working with.

Programming with this language was easy and quick thanks to MSDN Library also [14], freely available on the Internet, which collects all the documentation of the .Net Framework in a similar effective way as Java does. Thanks to the MONO development project [15], the project can be extended to other platforms based on Mac OS X or Unix systems. As future work, the team of students is in charge of extending .Net Framework to both Operating Systems.

In LSED laboratory the generated open-source DLL is used for remotely controlling a robot (based on an open-hardware Arduino platform), using a PC and the Wiimote, although the DLL can be easily integrated in any other C, C++, Java or C# project.

VI. HARDWARE DEMONSTRATOR

To demonstrate the capabilities of the gesture recognition system, students decided to show DLL’s functionality by

controlling a small basic robot based on an Arduino Duemilanova board [16]. This board is based on Atmega328 with 16KB to 32KB of Flash memory and clock speed of 16 MHz. Its physical structure was made from various recycled plastic parts taken from old or broken devices (Figure 4.).

The main feature of this system is offering a fully open-source-based hardware platform. The available development kit is completely free and all the reference related to its programming tips can be found on Internet. The main advantage when programming an Arduino board is the use of a high-level language, very similar to C: all the libraries needed are already included in the development kit.



Figure 4. The robotic platform controlled by the gesture recogniser

For demonstration purposes, students established a serial communication directly to the robot, so they could send the processed data from the recognition software in order to control the robot at will. In this case they simply recovered the data sent through the serial line and with the recovered command from the data they made the robot react in a certain way (to move forward or backwards or to turn on the left or on the right).

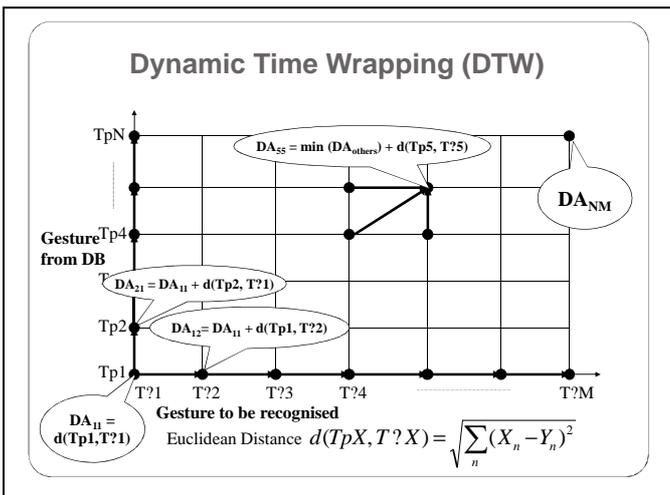


Figure 5. Graphical procedure of Dynamic Time Warping

VII. CONCLUSIONS

A new engaging project has been proposed, designed, implemented and tested for a PBL course on microprocessors. The use of cheap wireless game-oriented commercial hardware, open-source software and hardware and pattern-matching techniques is very attractive and has already engaged several teams of students during the last two years in a third-year microprocessor-based laboratory course. This freeware project is able to foster students' interest on developing innovative potentially-marketable systems.

The gesture recognizer estimates the movements of the user's Wiimote and by comparing the detected trajectory with the previously learnt movements, is able to carry out the associated actions in interactive applications (from robotics to virtual reality and multimedia presentations). Implemented as a configurable publicly-available library, it can be adapted to the needs of any course or student. In LSED laboratory at UPM, this open-source DLL is used for remotely-controlling a robot using a PC and a Wiimote. The developed DLL can be integrated in any C, C++, Java or C# project and a GUI application is also provided as a template for new applications or just for debugging purposes. Although the developed application only takes data from the accelerometers, data from the infrared camera and the Wiimote buttons is also available.

From a personal point of view, on behalf from the authors; the development of this kind of project based on the PBL methodology has been a quite profiting experience for both the students and the professor, although working with a different development environment and platform can imply an extra effort for the people involved, due to the fact that learning how to handle the different required tools takes time.

On the other hand, conducting an innovative and different project has increased people's awareness for this type of approaches. With this type of methodology we are capable of reaching others students and professors related with similar projects.

After this second-year experience, next year we can offer a standardized project to most of the students in the LSED laboratory course. There is a video demonstration at <http://www.youtube.com/watch?v=joo3G3BejwY>.

ACKNOWLEDGMENT

Authors thank J. Macías, R. San Segundo and all the members of the Departamento de Ingeniería Electrónica and the GRIDS innovation group at ETSIT-UPM, for their continuous effort to offer high-quality education to the students and for all their fruitful comments and suggestions that made this work possible. Authors also thank the IEEE Spanish Student Branch members at UPM for their support and also UPM innovation project "Implantación y evaluación de la metodología docente asociada a una plataforma no presencial de diseño electrónico orientada al autoaprendizaje" for partially financing this work.

REFERENCES

- [1] Buck Institute of Education., Project Based Learning Handbook, pp. 3-10, 2007.

- [2] H.S. Barrows, "Problem-based learning in medicine and beyond: A brief overview" in Bringing problem-based learning to higher education: Theory and practice, pp. 3-12. San Francisco: Josey-Bass, 1996.
- [3] G. Solomon, "Project-Based Learning: A Primer", in Technology and Learning, 23, pp. 20-27, 2003.
- [4] M. A. Albanese, S. Mitchell, "Problem-based learning: A review of literature on its outcomes and implementation issues", in Academic Medicine, 68, pp. 52-81, 1993.
- [5] Robert S. D'Intino, K. Weaver, Schoen Mark, J. Edward, Integrating a Project Based Learning Model into the Entrepreneurial University. NJ: Rowan University, 2007
- [6] J. Macías-Guarasa, J.M. Montero, R. San-Segundo, A. Araujo, O. Nieto-Taladriz, "A Project-Based Learning Approach to Design an Electronic Systems Curricula", in IEEE Transactions on Education Vol. 49, pp. 389-397, 2006.
- [7] S.A. Ambrose, C.H. Amon, "Systematic design of a first-year mechanical engineering course at Carnegie-Mellon University", in Journal of Engineering Education, 86, pp. 173-182, 1997.
- [8] G. R. Ryser, J.E. Beeler., C.M. McKenzie, "Effects of a Computer-Supported International Learning Environment (CSILE) on students' self-concept, self-regulatory behavior, and critical thinking ability", in Journal of Educational Computing Research, 13, pp. 375-385, 1995.
- [9] J. W. Pellegrino, D. Hickey, A. Heath, K. Rewey, N.J. Vye., Assessing the outcomes of an innovative instructional program: The 1990-1991 implementation of the "Adventures of Jasper Woodbury". Nashville, TN: Vanderbilt University, Learning Technology Center, 1992.
- [10] Microsoft Developer Network C# <http://msdn.microsoft.com/en-us/vcsharp/default.aspx>
- [11] John Sharp, Microsoft Visual C# 2005 Step by Step, Microsoft Press, 2005
- [12] Wiimote Library - .NET managed library for the nintendo Wii remote <http://www.codeplex.com/WiimoteLib>
- [13] A simple C# library for graph plotting <http://www.codeproject.com/KB/miscctrl/GraphPlotting.aspx>
- [14] MSDN library documentation <http://msdn.microsoft.com/en-gb/library/default.aspx>
- [15] MONO project http://www.mono-project.com/Main_Page
- [16] Arduino platform <http://www.arduino.cc/>

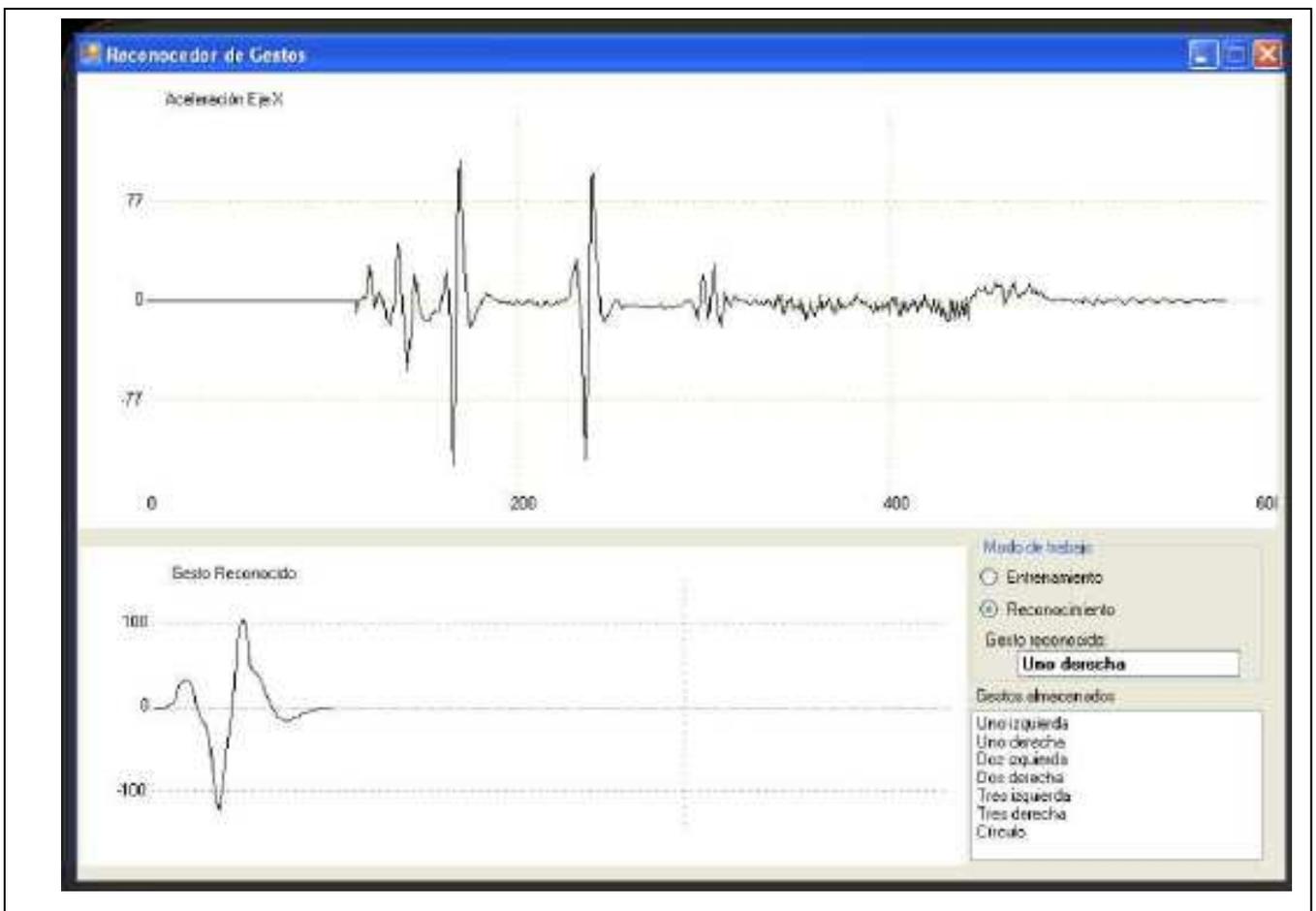


Figure 6. Graphical user interface of the developed gesture recognition system