

Internet-Based Teaching Evolution in Computer Architecture

Eugenio López, Elio Sancristobal, Sergio Martín, Gabriel Díaz, Manuel Castro, Juan Peire,
José M. Gómez and Paloma López

eugeniolopezaldea@gmail.com, {elio, smartin, gdiaz, mcastro, jpeire}@ieec.uned.es, jmgomez@iti-to.uclm.es,
plaldea@indra.es

Abstract - The DIEEC (Electrical and Computer Engineering Department) of UNED (Spanish University for Distance Education) in Spain is elaborating different tools to teach in Computer Architecture subject by using new technologies. DIPSEIL (Distributed Internet-based Performance Support Environment for Individualized Learning) is one of the projects where we are experimenting with teaching by task in Internet. We present this project and their advantages. By the other side we are elaborating lots of resources and evaluating them to incorporate Virtual and Remote Labs integrated in Learning Management Systems (LMS) or Educational Platforms. Lots of the decisions of this Web-Labs are the suggestions that students suggest in the DIPSEIL Architecture course.

Index Terms – Computer Architecture, Dipseil, iLab, LMS, Microprocessor Simulator, Virtual Lab, Remote Lab, Web Lab.

INTRODUCTION

In the new millennium where omnipresence in computer is evident, it is necessary to adapt the form to teach in computer architecture by using new technologies. The learning based in task was our bet and distance education was solved using Internet as tool to approach to all the students. In this article, we pretend to expose all knowledge and share our research that our Department [1] are acquiring and developing in Computer Architecture learning by Internet.

DIPSEIL

DIPSEIL is an on-line multilingual environment computer focused on education based on task (Figure 1) [2]. It is the continuation of the IPSS-EE (Internet-based Performance Support Systems with Educational Elements) project where teaching based in task was carried out to Internet following Van Merriënboer's four components instructional model, 1997, that support the learning of task designed to simulate real situations of work by computer. It was a two-year project within the European program Socrates/Minerva and the partners are from six European universities and one college. DIPSEIL has the next fundamental components:

tasks, support information, process information and step by step task practices.

Remembering, at the beginning of the IPSS_EE platform (old platform based in task) we probed a course called VHDL Simulation. The experience was a success so that we went on working hard in this way. The teaching based on task advanced with DIPSEIL, where we have developed a powerful course about Simulation and Architecture of Microprocessors.

DIPSEIL is a new concept that explains contents which can be organized based on tasks using new webs technologies. In this case, instead of elaborating the courses in the same server (IPSS_EE), DIPSEIL is distributed in five servers in different universities from Europe. The five servers that collaborated in the project, in different parts of the World were Spain, France, Ireland, Netherlands and Bulgaria. It is possible to select the language and the information of the user interface is divided in two parts, the information of the main server in Bulgaria (information of all the courses and general interface) and the content of the courses hosted in each server. As we wrote, DIPSEIL is similar to IPSS_EE but instead of working with only one server, we distributed the servers and information in each country, with the aim to obtain more speed. The software used is mainly open-based implemented in PHP and MySQL.



FIGURE 1
PRINCIPAL WEB-PAGE OF DIPSEIL

DIPSEIL has several advantages than IPSS_EE thanks to the experience of this one. With DIPSEIL, teachers can edit their own courses directly by Internet without need the administrator/developer. About this, each teacher can create the contents of the courses based on task and include them in the DIPSEIL platform in each place over the world. In these courses it is possible to add any learning objects and all sort of resources in lots of formats, not only in PDF and create contents in HTML format with the editor included. The main objective is to present the information in the form that the teacher considers more efficiency with all technologies available.

MICROPROCESSOR SIMULATOR COURSE IN DIPSEIL

DIPSEIL platform helped us to improve the teaching in Architecture Microprocessor subject with the creation of one course dedicated to practice with a simulator. The course was created with twelve tasks that cover the fundamental issues of one elemental course, two practices test, lots of support materials, tutorials and links. To do practices with a real microprocessor the students used free Easy68k [3] as Microprocessor Simulator (Figure 2). The first experimental course was a success and we are repeating this year with more experience acquired. In all the courses we always do two groups to compare students that elaborate the course in classic study and students that are using this Internet Tool. In this way, we can know if this tool can improve in reality, the quality of education in this area. The results and comparatives of the different courses are demonstrating that the students that use Computer Architecture course in DIPSEIL have better results in the final note and they are better training in the solve the practices of the usual course. Other intention of the course is prepare (pre-laboratory time) in order to the laboratory time will more useful.

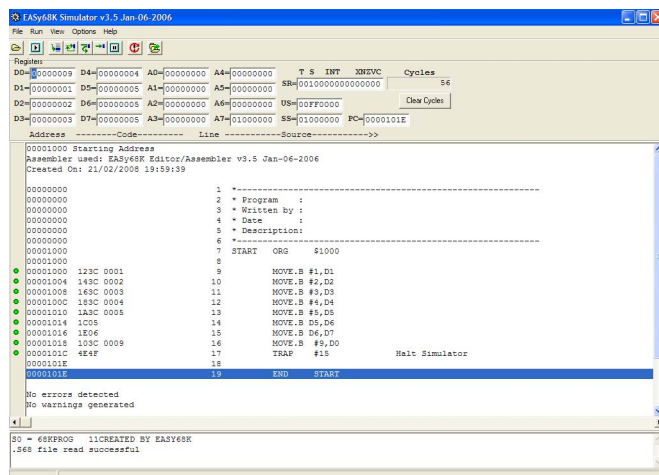


FIGURE 2
EASY68K MICROPROCESSOR SIMULATOR PROGRAM

At the same time, we are doing surveys and evaluations about the use of this Microprocessor Simulator extracting

information about how can we do the new Virtual and Remote Microprocessor Lab integrated, in the future, in LMS or iLab. We are studying the real needs of students adapting the new tools used in the practices.

The main steps that we have carried out in order to obtain the correct results to evaluate this course are summarizing in the Figure 3. In general, in our Microprocessor Course, the students elaborated one questionnaire at the beginning about learning styles, and then, when they finished technical knowledge they filled a reflexive one to evaluate the environment DIPSEIL. We included a third survey to evaluate the simulator used and sound out how a simulator must be to get our educational objectives in Microprocessor Web-Labs research.

MICROPROCESSOR COURSE'S EVOLUTION

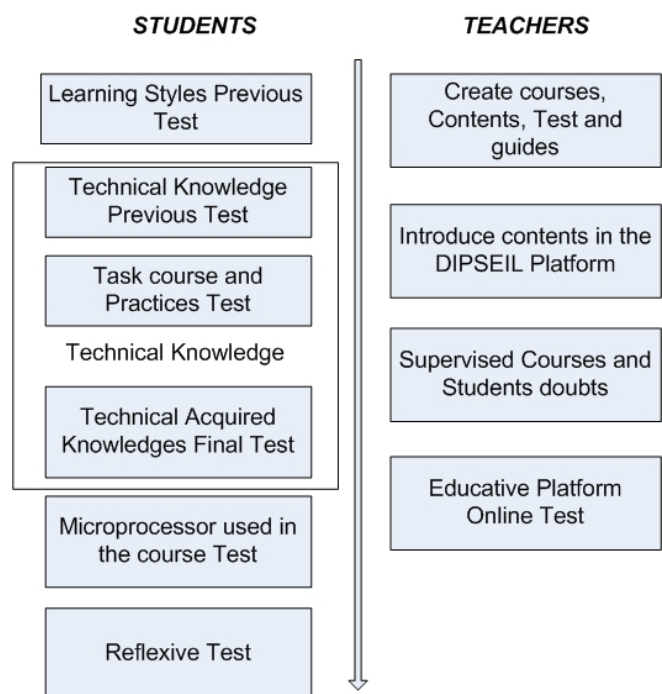


FIGURE 3
STEPS CARRIED OUT BY STUDENTS AND TEACHERS IN DIPSEIL PROJECT

Coming back to description of the course, we developed twelve task dealing each part of the subject in order to students could get practice before the laboratory time. Each task contents Task Specific Training, Instructions how to Perform, References, Expert Advices, Communication with the tutor and software to practice. The student begins the course reading a general guide to access the course and then they read the first task. In that moment, they have all resources available and must deduce how to use the knowledge to manage or pass the task.

The general points of each task are:

- **Task Specific Training:** students can download information about the task that they are studying. This

point is used to guide in the way to solve the task in short.

- **Instructions how to perform:** in this point students can find general information about the subject. All formats are possible.
- **References:** The teacher can include URL and books references and change these during the course because it is possible that the URL changes throughout the time.
- **Expert Advices:** In this section are included quick questions and answers where students can solve doubts like frequency answers questions.
- **Communication with the tutor:** where students can send information to the teacher and send an e-mail.
- **Software Simulation:** Students can execute the software that they need to solve the task.

There were a lot of students registered in our Department in Electronic and Architecture Computer. We encouraged students to collaborate in the course, students who joined in this course are obtaining better results in the final note than students that continued the course without experiment. But we are doing more experiences to contrast this information doing new courses.

EXPERIENCES OF THE COMPUTER ARCHITECTURE'S COURSE

With the experiment in the Architecture of the Microprocessor course, we studied several aspects that can improve it. For this, we elaborated a monographic to introduce technical knowledge in a logic form from the system to elements of computer looking for the need of each element of microprocessors (Figure 4). These contents have been included in DIPSEIL at the beginning of the course before the task. We assure like that students will know what they are doing and are oriented in general in their task.

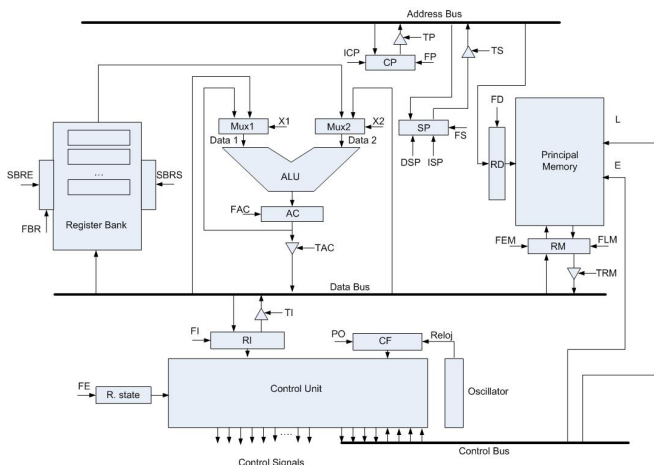


FIGURE 4

FINAL MICROPROCESSOR EXPLAINED IN THE NEW MONOGRAPHIC INCLUDED IN THE COURSE OF MICROPROCESSOR (DIPSEIL)

The simulator that we chose to practice in the course was Easy68k Simulator Program. This simulator is easy to

understand and it has a graphic interface very usable. It is free software and students can download by Internet. But our real proposal is improve our own software studying reactions of the students with the courses using Easy68k (Table 1) and mixing with the experience of the professionals in this subject. We elaborated a survey after the course to the students oriented to know what aspects of the simulation had to be room for improvement. In the table 1, we summarize several sections of this survey. In the last question of this one, students could express general points of their ideal simulator.

In the Table 2, it is showed the data for each student and the commentaries over the relevant aspects of the simulator used as well as it was added the different suggestions that students had proposed.

TABLE I
SURVEY'S GENERAL DATA

Items	Data
Students that did the course of Microprocessor Simulation and presented final report	15
Students that did the questionnaire of the Simulation in Microprocessor	10
Averaged time used in the answers	26,4 minutes
Sort of Microprocessor used in the course	MC68000
Simulator used in the course	Easy68 v3.5 and v3.6 Beta
Students that prefer to work <i>off-line</i>	6
Students that prefer to work <i>off-line</i>	4

EVOLUTION OF RESEARCH IN MICROPROCESSOR SIMULATOR

Almost fifteen years ago, in 1994, we began to create new microprocessor simulators and investigate the best way to teach its architecture. We developed the first Simulator with the aim to understand the work of MC6800 that has been the micro most representative. It was born a project that marks an important line to start the didactic evolution in this subject, Simulation of Microprocessor with the aim to develop the learning distance education and to reinforce the traditional one. Suddenly this simulator was improved and it was developed another by the Intel 8085. These applications were programmed in Turbo Pascal and based on DOS. Then, in 1995, the Department achieved a CICYT Project in which was developed several researches about design and creation of different tools for the simulation of advance microprocessors. It was pretended with success, GUI MS-Windows support, hardware configuration editor, intelligent assembly program editor, multimedia characteristics and work with 16 or 32 bits.

FUTURE MICROPROCESSOR WEB-LAB

Continuing with the research of the Microprocessor Simulators the Department's trying to make good use of the potential of the LMS – Learning Management System [4] [5] and the services that offer the iLab Server Broker in the

Session T2B

Virtual and Remote Labs (iLab) [6]. We are elaborating a Web-Space where the Word “reusable” has the top sense. We desire to develop a Web-Lab with LMS’s resources management and iLabs Services integrated without reprogramming all the services in each Web-Lab. Several educational platforms that contain e-learning standards (for example Scorm, Ims [7] [8]) provide services as authentication, forums, chats, schedule management between other that we can use in the Microprocessor Web-Lab and use in general in each Web-Lab.

TABLE II
DATA OF THE SURVEY OF MICROPROCESSOR'S SIMULATOR USED

Time	Prefer on-line/off-line	Commentaries
20 m	Off-line	Easy to use. It is useful to see registers and state bits directly in the monitor.
15 m	Off-line/ On-line	Possibility of executing step by step the assembler instructions. He proposes a multidocument interface.
6 m	Off-line	Let the values of registers and memory state and the pile pointer.
30 m	On-line/ Off-line	He/She suggests that Editor/Compiler and the Simulator will be in only one application. He/She proposes a debugger in real time.
20 m	Off-line	He/She suggests improving the debugger indicating exactly where the error is and what kind is. When it is executed step by step it would be good to illuminate the different bits of memory affected. He/She considers useful to work with several windows.
20 m	Off-line	He/She suggest a help in the origin language and to see the chronograms that show the process.
35 m	On-line/ Off-line	he/she suggest that all operative systems can be execute the software simulator and incorporate an instructions dictionary in real time to help in the edition of the assembler program.
15 m	Off-line	He/She considers that executes an instruction to instruction the code and analyzes the state of registers and memory. The graphic interface is very intuitive.
45 m	On-line/ Off-line	He/She consider useful the information of the register and the memory next to the possibility to execute the program instruction to instruction. He/She consider useful the hardware sights and break points.
57 m	Off-line	He/She is the only student in utilize the Easy68k Editor/Assembler v3.6 beta. He/She considers intuitive to manage. He/She suggest incorporating help in his/her own language. He/She doesn't like the memory interface.

In this way it is possible to minimize the effort and develop of management for remotes and virtual labs users and developers allowing multiple universities with different nets share the material of each Web-Lab in any part of the world and provide one set of services and tools. It is understand the magnanimous possibilities of this way to work and by this, the actual performance of development in this area by the Department.

With the idea of communicate LMS, iLab and Web-Lab we are defining an communication interface or middleware (Figure 5). In this way the developer of the Web-Lab won't pay attention of the use of the LMS or iLab programming.

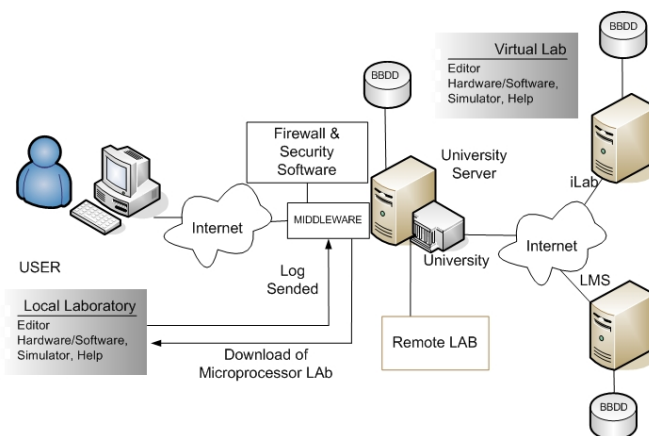


FIGURE 5
ENTITIES' DISTRIBUTION IN THE MICROPROCESSOR VIRTUAL
AND REMOTE WEB-LAB PROJECT.

The Web-Lab will communicate with the Middleware developed to reuse the complex resources of the LMS or iLab giving in the application of the Virtual Lab the task that only are competence of this one. Nevertheless, the Web-Lab will have own existence endowing with the minimal management services.

With the suggestions of the students commented before, we pretend that the user can work with or without connexion to Internet. In this way, users can use the laboratory without dependence of this one. When the students use the laboratory without connexion, it will create a log and properties folder with the necessary information for identification and evaluation subsequent. Later, if it is desired, the students can upload this information in the Web-Lab to evaluate. In this way, it is possible to work in asynchronous time and no dependence of the Internet but using later all the advantages that the connexion can offer us.

Following the description we details the principal points that Microprocessor Web-Lab must provide extracted of the knowledge of the professionals and the suggestions and needs that students expressed in the surveys carried out in the Dipseil Microprocessor's course.

First, it is defined the services that Web-Lab must offer in local work:

- Editor/Debugger
- Compiler
- Simulator
- Hardware Configuration
- User Interface
- Reports generation (Calculus sheet, xml, pdf or text archive)
- Properties and log folder
- Offline help

The generation of the properties archive consists in one xml where it is documented the activity of the user and archive error .log where to document the errors that can be produced during the execution of the application, letting to have a control of the possible bugs.

The properties archive includes the next information:

- Student that works in the laboratory
- Time dedicated
- Activities (screens that he see and times that use each resource in order to analyze the laboratory and improve it)
- Title of the programs created

Next, we expose the services of the laboratory in connexion to Internet. Then, this services are the same that the local work but adding:

- Dynamic Help
- Collaborative Tools
- Education management

The real difference with offline work is in the collaborative tools, dynamic help and the education management that permit to use the resources of the LMS and iLabs that we expose before.

The services that the laboratory will provide and it will be used basically by the middleware are:

- Authentication
- Chat and dynamic class
- Forum of microprocessor
- Schedule management
- Repository of archives and programs
- Test generation and execution

The normal use of the user in the Microprocessor Simulator Laboratory is the edition of the assembler program, compilation, hardware configuration, simulation and visualization of the results by monitor (Figure 6).

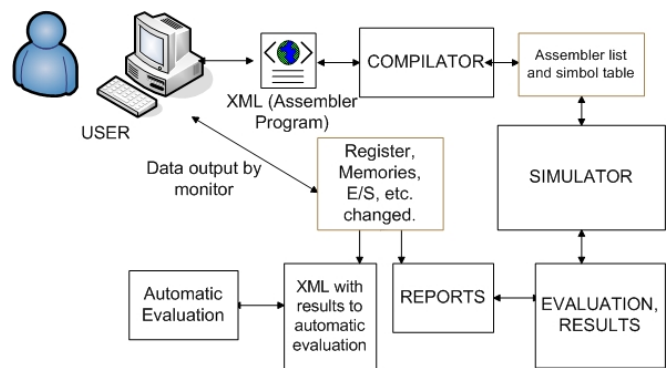


FIGURE 6
NORMAL USE OF THE MICROPROCESSOR'S SIMULATOR LABORATORY

The editor will generate one xml archive according to data input and receive data of the local repository or write the assembler program.. Before generate the xml, the editor will do the syntactic analysis to debug the errors and indicate where and which one in the program. The compiler will have as input xml archive generated in the editor and will develop an assembler listed and its symbol table to output. The simulator will read the dates of the symbol table and the assembler list to change the data of the registers, memories and E/S of the microprocessor selected in its hardware configuration. The user interface will represent by monitor the hardware selected as memories, registers, E/S and peripherals and the microprocessor principal blocks as ALU, UCP, data path in different abstraction levels.

In the figure 7 it is summarizing all the activity of the integrated laboratory in the services that different platforms offer us.

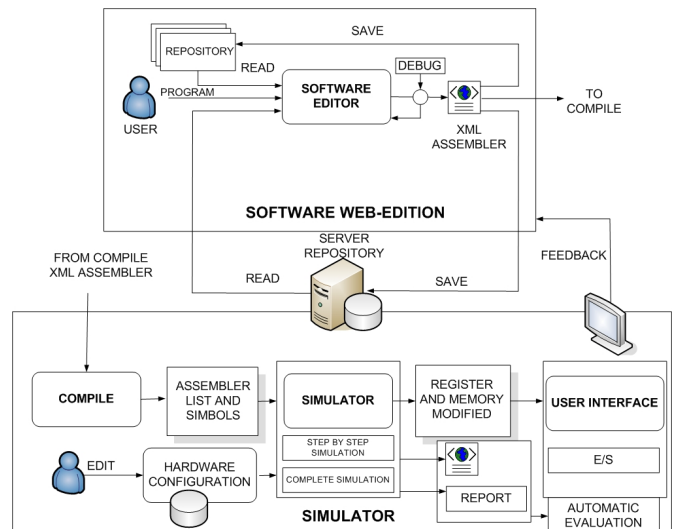


FIGURE 7
COMPLEX ACTIVITY OF THE MICROPROCESSOR WEB-LAB

By other side, we are studying in how present the graphical user interface in any microprocessor laboratory. For this, we are defining a class library in java [9] [10] [11] [12] That permits to organize the graphic interface of any microprocessor in a different levels of hardware. In this way, the creation of the new microprocessor simulator won't imply that the developer must to reprogram all the aspects that we consider similar in all microprocessor simulators. At the same time, if we can improve this interface we only must to change the library.

SUMMARY AND CONCLUSIONS

All the Microprocessor Learning Research have given important knowledge that we are making good use to set out the way of learning of the future by using new technologies. In this moment, we are doing Projects and Research Doctoral where the learning of the microprocessor is having a new dimension. One of these future lines is the cited Learning and Simulation using tasks based on-line learning. On the other side, in the developing simulation area we are working to implement a new application based on simulation tool written in Java and totally web-based being integrated inside the Learning Management Systems as well as has a stand-alone version. Each part of the program will be modular and we can use like an object learning per each educational platform that carries out with the standard. All the programs will be developed with free-software and they will content database instruction management and multi-level simulation help. The geographical distance loses increasingly sense with the utilization of these tools. Educational platforms across Internet like DIPSEIL, multimedia and possibility of improving and extending the contents in real time in these tools together, the develop of the standard web-based microprocessor software system simulator where includes all analysis process of evaluation and control education are, among others, great possibilities that it is offered to the education system. Respect to the analysis done in the courses based on task in DIPSEIL and the test, we can conclude several things in order to improve the future design of the Virtual Laboratory in Microprocessor's Simulation. At first we are designing to provide a connection with one educative platform or web-space where students and teachers can use the services that it offers: authentication, test generation, times used, forums or other general services. The simulator, all by Internet, must provide an editor to the creation of programs, where it includes help in real time as general help on tutorial and use mode. The editor should be compiler or select the assembler sentences with data base. It's important to show state register, memory data and all hardware blocks in different level of detail. They should be an option to visualize the chronograms. The micro will be accompanied with several peripherals and their E/S. All must be intuitive, comfortable to manage and always with help resources. We expect with all this new developments to improve the quality of learning in this area but always having in count the opinion and suggestions of the students.

REFERENCES

- [1] DIEEC Department (2008). <http://www.ieec.uned.es/> Accessed on March 2008.
- [2] DIPSEIL (2008). <http://www.dipseil.net/>. Accessed on March 2008.
- [3] Easy68k. <http://www.monroeccc.edu/ckelly/EASY68k.htm> Accessed on March 2008.
- [4] dotLRN. Learn, Research, Network. <http://dotlrn.org>. Accessed on March 2008.
- [5] Moodle. <http://moodle.org> . Accessed on March 2008.
- [6] iLab. Internet Access to Real Lab (MIT). <http://icampus.mit.edu/ilabs/> Accessed on March 2008.
- [7] Adlnet. <http://www.adlnet.gov/scorm>. Accessed on March 2008.
- [8] Ims. <http://www.imsglobal.org/question/> Accessed on March 2008.
- [9] Ceballos F. J. JavaTM 2. Interfaces gráficas y aplicaciones para Internet. Ed. Ra-Ma (2006).
- [10] Castro M. and other. Estructura y Tecnología de Computadores I. Ed. UNED (2002).
- [11] Froufe. A. Java Server Pages. Ed. Ra-Ma (2002).
- [12] Castro, M.: Laboratorios Virtuales para Enseñanza en Internet – Situación y Desarrollo. I Jornadas sobre e-Learning (TEL2005). Madrid, 2005.

AUTHOR INFORMATION

Eugenio López is with Niedax Kleinhuis Company, eugeniolopezaldea@gmail.com.

Elio Sancristóbal is with the Electrical and Computer Engineering Department, UNED (Spanish University for Distance Education), elio@ieec.uned.es.

Sergio Martín is with the Electrical and Computer Engineering Department, UNED (Spanish University for Distance Education), smartin@ieec.uned.es.

Gabriel Díaz is with the Electrical and Computer Engineering Department, UNED (Spanish University for Distance Education), gdiaz@ieec.uned.es.

Manuel Castro is with the Electrical and Computer Engineering Department, UNED (Spanish University for Distance Education), mcastro@ieec.uned.es.

Juan Peire is with the Electrical and Computer Engineering Department, UNED (Spanish University for Distance Education), jpeire@ieec.uned.es.

José M. Gómez is with Electrical, Electronic, Control and Communication Engineering Department UCLM (Castilla-La Mancha University), jmgomez@iti-to.uclm.es

Paloma López is with Indra Sistemas S.A. company, plalde@indra.es