

# Developing Signal Processing Applications using MultiPAS in Accordance with the Bologna Requirements\*

BEGOÑA GARCÍA ZAPIRAIN, AMAIA MÉNDEZ ZORRILLA, IBON RUIZ OLEAGORDIA and JAVIER VICENTE SÁEZ

University of Deusto. Avda. Universidades 24, 48007 Bilbao. Spain. E-mails: mbgarciazapi@deusto.es, amaia.mendez@deusto.es, ibruiz@deusto.es, jvicente@deusto.es

*This proposal introduces and evaluates a dynamic system of teaching signal processing using the Application Programming Interfaces (APIs) called MultiPAS. The authors are engaged in the Bologna process; thus, this subject is defined following the European Credit Transfer System (ECTS). The method consists of developing new signal processing applications using the main programming languages (Java, C++ and C#) and connecting them with 'Octave' mathematical software by means of the three specific gateways developed by the authors. It is important to stress the free-software nature of the developed gateways: as no licence is necessary, student access to this program of scientific calculus is easy. The learning results of this pedagogical innovation have been validated over the last three years.*

**Keywords:** signal processing teaching; engineering education; Octave; Programming Gateway; European Credit Transfer System

## 1. BACKGROUND

EUROPEAN UNIVERSITIES are currently involved in the European convergence process, which implies important changes in regard to the Bologna Accord (1999) [1, 2]. Universities are making changes in their programmes in order to favour student mobility and obtain harmonised academic credentials. Moreover, all of these changes involve a new teaching pedagogy. To facilitate this process, a common framework for comparing studies within the EU has been defined: the European Credit Transfer System (ECTS).

In this system, each subject has been assigned an ECTS equivalent to the number of hours that students dedicate to their studies. According to this system, these hours include teaching hours as well as time devoted to studying, exercises, assignments or exams.

The authors have applied the ECTS [3] measurement to the signal processing subject programme studied in telecommunications and electronic engineering courses. The new system has implied several changes: on the one hand, the number of hours dedicated to each activity has changed and, on the other hand, pedagogical innovation elements have been inserted to replace the traditional method of master classes.

In order to do this, the authors have developed an application called MultiPAS, which serves as a multibridge between different high-level languages

and the scientific program Octave (equivalent to Matlab). The students using MultiPAS belong to different engineering discipline and are accustomed to programming in different languages. Therefore, the authors added a number of elements to signal processing learning so that the student learns different algorithms but also has to program them from various familiar program frameworks.

For example, electronic engineering students commonly work in C++, and telecommunications engineering students are acquainted with Java. A real application development with signal processing algorithms is demanded from both. As a result of MultiPAS development, both groups of students can use the programming language learned and include the commands to be executed by Octave. This is one of the main goals of this tool.

The primary aim of the work presented here is to improve the signal processing learning process while respecting the Bologna requirements and including pedagogical innovation. To this end, we have made a tool available that enables the rapid development of applications, with a user interface using signal processing algorithms implemented with other tools (Octave). This being the main aim, a number of secondary objectives can also be achieved:

1. Planning a new signal processing subject in engineering degrees in accordance with the Bologna agreement requirements: the student's class work, as well as that out of the class, will have to be taken into serious consideration.

\* Accepted 3 May 2010.

2. Facilitating the development projects and graphics environments in various programming languages (Java, C++ and C#) with straightforward access to mathematical functions through the Octave/Matlab tool. This will enable engineering students with a command of high-level languages to carry out creative projects and overcome the difficulties involved in having to develop graphics environments with Octave/Matlab, thus leaving this mathematical tool for signal processing and employing the languages known for the rest of their project's functions.
3. To publish the project at Sourceforge.net for use and assessment by the scientific community.

This paper is divided into the following principal sections: Section 2 describes the current proposed methods, Section 3 presents the authors' methodology design, and Sections 4 and 5 report on the results, conclusions and future work.

## 2. METHODS

The didactic and technical methods used in the development of this project are described below.

### 2.1 Classical Method

The traditional lecture (probably the oldest teaching method), tutorial and practical (L/T/P) method of teaching digital signal processing organises the student's education into two blocks: one theoretical (four hours a week) and the other practical (two hours a week); both are based on the same textbooks [4, 5].

At first, students learn the theoretical basics related to each topic where the mathematical and analytical descriptions are produced.

To improve students' comprehension and assimilation of information, the following reinforcement techniques are used:

1. Exercises explained in class
2. Homework.

Once students pass the theoretical study stage, they proceed to the laboratory to have a guided practical session using simulation tools, such as Matlab/Octave/Simulink, and programming algorithms in a hardware system based on a Digital Signal Processor (DSP).

Students have one week to hand in a report on the experiments, which should include both the experimental results and the theoretical developments justifying those results. The final mark is determined considering the result of the theoretical exam as well as assessment of the reports on the experiments carried out in the laboratory.

Class attendance is obligatory. There is much evidence that examination performance improves with increased rates of class attendance [1–3]. However, Hyde *et al.* [4] cautioned against a

mandatory attendance policy, having found from their research that a sizeable group of students did very well in learning the required course material without the aid of lectures.

It is very difficult to work on and assess student motivation with the traditional method given the high number of theoretical class hours. On many occasions the students do not get to observe the applicability of what is explained in class in the short run. Practical sessions are limited and strictly controlled time wise, and this lack of flexibility prevents the professor from favouring autonomy and creativity on the part of the student, factors that are highly valued in future engineers. Moreover, the classic lecture-based method is not always compatible with the competence-based form of teaching required by the EHEA.

### 2.2 Proposed Method

The first step taken was to apply the ECTS [6] credit system to the subject: 7.5 ECTS credits currently go to the subject, which implies a total of 175 hours. The distribution of these hours in accordance with the estimated work times is:

Class work: 78 class hours + 4 hours of exams  
Work outside the classroom: 93 hours

The course will last for 13 weeks of class. Students will attend four theoretical classes and 2 hours of practical laboratory work per week under the supervision of a professor. This makes a total of  $4 \times 13 = 52$  theoretical hours and  $2 \times 13 = 26$  practical hours: a grand total of 78 hours of class work, during which the following activities will be carried out:

1. Attending lectures: 25 hours
2. Attending practical classes: 24 hours
3. Classroom exposition: 12 hours
4. Solving problems: 17 hours
5. Taking a theoretical exam: 2 hours
6. Taking a practical exam: 2 hours.

The 93 hours of the student's independent work must be devoted to

1. Reading and studying the topic: 30 hours
2. Exercises: 38 hours
3. Work time for experiments: 25 hours.

The teaching-learning strategy will use two different methodologies:

1. Lectures, in which the professor gives details of the subject's technical content
2. PBL (project-based learning), in which the students do group work on several real projects that oblige them to use signal processing algorithms [14–16].

This methodology follows the teaching-learning model based on competences, as in, 'the network of behaviours organised at the core of a mental structure, also organised and relatively stable and movable when necessary' [12]. This implies that the study approach is oriented toward providing

students with the ability to deal with real and complex situations, for which they will need to assimilate knowledge, skills, competences, attitudes and values. As for the teaching of this subject, the professional profiles of each degree were taken as a starting point, in accordance with the Spanish Ministry of Science and Innovation and professional colleges.

Competencies [12] that are common to the degrees of telecommunications engineering and electronic engineering were chosen, such as designing software for signal processors and digital filters, depending on the application, and establishing the requirements and specifications for signal processing signals and systems. This would be done through the strategy of theoretical explanation.

A number of specific competencies belonging to each degree were also selected and will be studied in accordance with the PBL strategy [17–20]. In the case of telecommunications, the specific competence to be worked on is that of analysing, simulating and developing algorithms for modulation/demodulation, detection and codification of signals. Electronic engineering students will develop the specific competence of designing signal processing algorithms for their implementation in hardware [13].

This new learning methodology will be assessed by the participation of two control groups. The first followed the traditional method explained in Section 2.1, and the second group followed the method proposed in this article (2.2).

### 2.3 Background to MultiPAS Gateways

The main technologies used in the development of the proposed APIs are detailed below.

#### 2.3.1 Basics of Octave

Octave [7] is a high-level language for numerical calculation whose syntax is compatible with Matlab, but is developed by the free software community.

The Matlab [24][25] and Octave calc engine languages are the same, the main differences being that not all the Matlab toolboxes are implemented in Octave. Nevertheless the signal processing and image processing toolboxes are 90% compatible, these toolboxes being the ones that motivated the development of MultiPAS. The objective is to use the signal and image processing toolboxes employing different programming languages.

#### *What makes Octave different from other programming languages?*

Octave is particularly oriented toward the scientific world. Among its main differences from other programming languages, the following stand out:

1. Native matrix operation
2. Native operation with complex numbers
3. Interpretation of language.

These characteristics mean that scientific algorithms can be developed in a far shorter time than in other programming languages. Therefore, Octave is the ideal language for the development of digital signal processing algorithms, digital image processing, control systems, statistics, etc.

MultiPAS is a group of parsers allowing the use of Octave calculus power using different languages (Java, C y C#), Whereas Scilab is a scientific calculus language. A series of gateways between these languages to Scilab, instead of to octave, could have been implemented. Scite is an alternative for Octave, not for MultiPAS.

Octave and Scilab are two programming languages for scientific calculus that have quite a lot in common. Both have a similar syntax and both are free software.

Scilab could have been chosen as the scientific calc engine, yet the authors preferred the Octave option for various reasons. The main motivation for choosing Octave rather than Scilab is that the language used by Octave is more widespread in the university context in which we work and also because the languages of Octave and Matlab are compatible.

#### *What are the disadvantages of Octave?*

Although Octave is an ideal language for the development of scientific applications, as they can be carried out in a short period of time, it has some drawbacks, one of which is linked to the speed of computation. Being an interpreted programming language, Octave is slower than a compilable language, because the latter generates native instructions for the processor, which requires less time.

The second disadvantage is related to the graphic environment. Applications with Octave are executed on a console with the single possibility of producing graphic data displays. This makes it impossible to develop user interfaces that allow the user to interact with the application.

#### 2.3.2 C++

C++ [8] is a programming language designed by Bjarne Stroustrup in the mid-1980s as an extension to the C programming language. C++ is regarded by many as the most powerful language because it allows the operator to work at both high and low levels. However, at the same time, it is the one that bears the least number of automations (as with C, almost everything has to be done manually), which makes it difficult to learn.

The following are some of its main characteristics:

- Object-oriented programming
- Portability
- Brevity
- Modular programming
- Speed: The code resulting from a compilation in C++ is very efficient thanks to its ability to perform as a high- or low-level language and also because of the reduced size of the language.

### 2.3.3 C#

C# (C Sharp) is the new general-use language designed by Microsoft for its .NET platform [9]. Its syntax and structure are very similar to that of C++. However, its straightforwardness and high degree of productivity are comparable to that of Visual Basic.

The following are some of this language's characteristics:

- Simple
- Modern
- Object-oriented
- Efficient.

### 2.3.4 Java

For scientific computation, there are languages that are more precise than Java [10] and that are capable of carrying out a wider variety of calculations. In addition, these languages operate natively with matrixes and complex numbers, whereas Java does not. Thus, languages like Matlab [11] and Octave are used in scientific computation. It is much more difficult to program scientific calculations in Java than in scientific languages, which is a great disadvantage for Java.

Even so, there are people who use Java to develop programs including scientific computation. It is necessary to spend much more time programming scientific calculations with Java than with scientific languages, but Java does have many options for the development of user interfaces. Free scientific languages such as Octave can be used only in the command line, and applications cannot be created with them.

## 3. DESIGN

This section describes the software system developed by the authors to carry out the PBL [21, 22] learning-teaching strategy in the area of signal processing, based upon the previously explained Bologna requirements. Students download the following information from the subject page:

- A general overview of the project to be developed
- A bridge system developed by the authors to connect high-level languages with Octave
- A user manual for these bridges with examples of commands to the connection functions.

Since the students come from two different categories of engineering, the bridge system was developed to connect the C++ language used by electronic engineering students as well as C# and Java [23], used by future telecommunications engineers, with algorithm-programming tools for signal treatment.

In order to obtain the credits corresponding to the group project, students have to program a final project that can be applied to a real-life situation with a user interface including the algorithm

programmed in Octave/Matlab through MultiPAS.

### *Example of Project: Noise Subtraction Project*

*The initial condition and given material:* Two microphones have been set up in the front and rear of a motorbike helmet. The microphones are connected to the computer of the laboratory, the front microphone being the input left channel and the rear one the right one. On the other hand, the loudspeaker is connected to a white noise source.

*What is required of students:* The project consists of capturing the microphone signals and implementing the white noise removing algorithm. The capture of the signal is implemented in programming high level and the captured samples are sent to Octave through MultiPAS. The developed adaptive filter algorithm must be implemented in Octave. The result must be saved in an audio file.

The assessment of the project is carried out taking into account two milestones and the final mark is obtained, making the mean of the marks scored:

1. The Signal to Noise Ratio of the all groups is measured using MDVP. In this way, the results can be classified according to the quality.
2. The audio subjective assessment carried out by the students is done.

Figure 2 summarises the methodology students must follow to complete successfully the PBL [26] learning strategy. The first step is to program the algorithm using the Octave mathematical tool. Once the correct functioning of the algorithm has been verified, each student moves on to develop the user interface using the high-level language pertaining to the degree he or she is pursuing.

Block 3 in Fig. 2 combines the specific learning in the area of signal processing (the processing of input signals and the generation of output signals) with the results displayed through the user interface. Figure 1 shows the communication between the screen where the end user introduces the data, the block running the algorithm that processes this data, and finally the production of the results by means of some output parameters that will be displayed on the user interface screen. As indicated in Fig. 1, MultiPAS permits the user to execute Octave code from other programming languages (C++, C# and Java), as can be seen in Fig. 2. The program core is developed in a source language such as C++, C# or Java, and the complex mathematical calculus is done by Octave.

The main work of these APIs is the exchanging of variables between the application interface (developed in the previously mentioned languages) and Octave. Thus, the design and the methodology needed to use these APIs are the same.

Communication between the bridge languages in each case is simply based on three main classes:

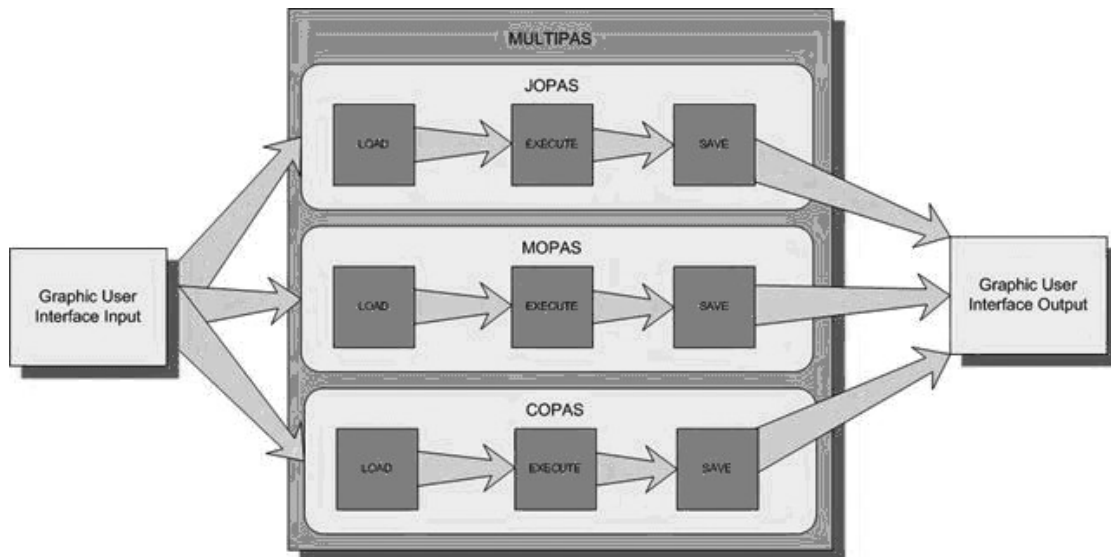


Fig. 1. Communication in MultiPAS.

1. 'ParserOctave' class. This is the main API class; it manages communication with Octave. It is responsible for loading the Octave thread and administering its input and output through the control of Windows API functions. The streams are redirected to input/output stream descriptors that send the Octave instructions ('execute' function) and obtain the results in the appropriate variable ('executeAndSave' function).
2. 'JoPAS', 'CoPAS' or 'MonoPAS' class. In each of these classes, the Parser, previously defined, is started and closed. Additionally, the Octave's algorithm is programmed with the corresponding execution form, saving the points of the X and Y axes.
3. 'GUI' class. This class is based on a plot. With the click of a button, the 'JoPAS', 'CoPAS' or 'MonoPAS' class is now launched. However, the real issue is to implement a correct library (depending on the language—Java, C++ or C#); the chart is represented using the X and Y axes.

Block 4 in Fig. 2 represents the final stage of assessing the system developed by the student. The system itself will enable him or her to generate reports on the test trials carried out, which should provide evidence that the whole system performs satisfactorily.

Figure 3 shows the user interface for the project on the theme of oesophageal voice analysis, including processing carried out by the students.

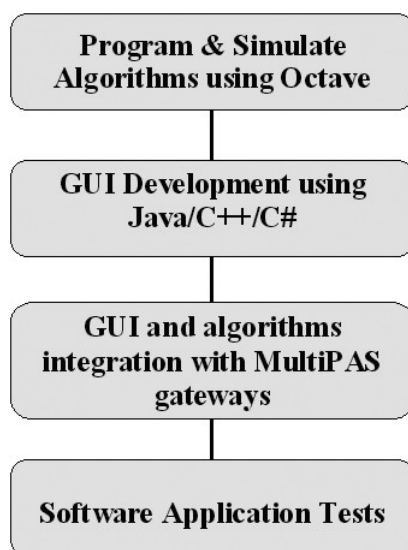


Fig. 2. Work methodology.

#### 4. RESULTS

The results obtained from this experience are explained below. In order to understand better the success of the proposed pedagogy, the authors distinguished two assessment areas: academic level and degree of student satisfaction.

##### 4.1 Academic Results

To evaluate the proposed method according to academic results, the final grades obtained by students following the traditional method in the years 2006 and 2007 were compared with those obtained with the proposed method in 2008. The student group is made up of all telecommunications and electronic engineering students. The total annual sampling of students taking part in this study was 53 in 2006, 51 in 2007 and 49 in the year 2008.

Table 1 shows the results obtained in the years 2006, 2007 and 2008, giving the number of students

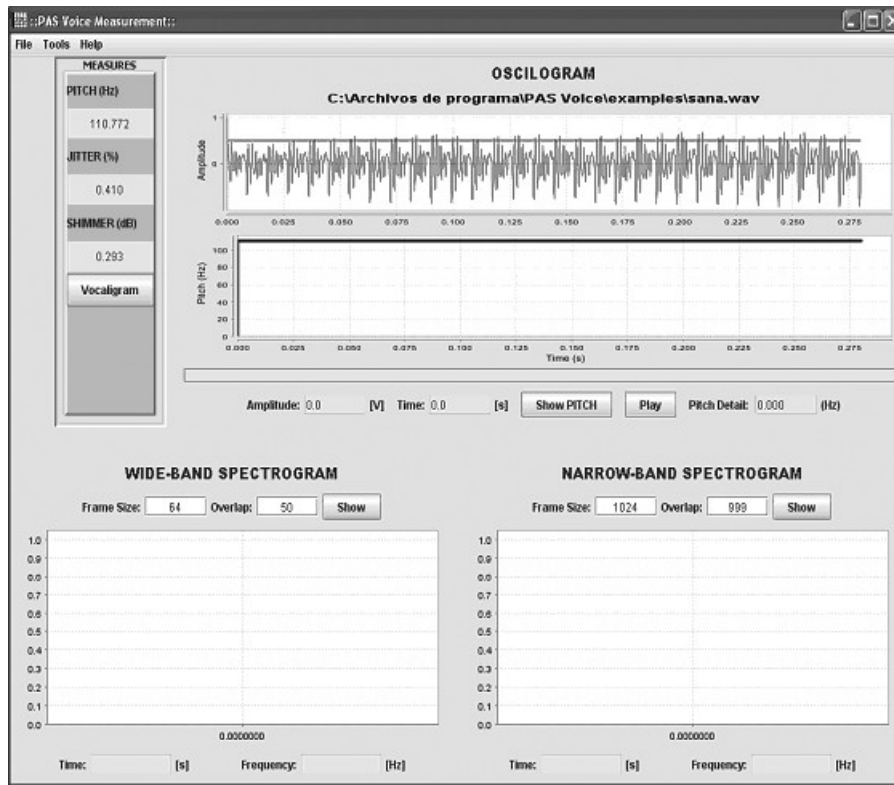


Fig. 3. Application example developed with MultiPAS.

Table 1. Students' marks

Mark	2006		2007		2008	
	Students	%	Students	%	Students	%
0-3	9	18.37	13	25.49	3	6.12
3-5	18	36.73	15	29.41	10	20.40
5-8	23	46.94	19	37.25	31	63.28
8-10	3	6.12	4	7.85	5	10.20

achieving scores between 0 and 3 out of 10, 3 and 5 out of 10, 5 and 8 out of 10, and 8 and 10 out of 10, respectively. For each year, the number of students whose marks were within each of these categories was counted, and the percentage corresponding to the total number of students being assessed was calculated.

The horizontal axis in Fig. 4 represents the four categories of marks obtained by the students, in accordance with the four previously explained ranges (0-3, 3-5, 5-8 and 8-10). Three columns appear within each range, each one corresponding to the three years under study. The vertical axis shows the percentage weighting of each value represented in relation to the sample group of students.

From the analysis of Fig. 4, it is plain that there was a significant improvement in the number of students who received a failing score in both the 0-3 range (18.37% in 2006, 25.49% in 2007 and only 6.12% in 2008) and the 3-5 range (36.73% in 2006, 29.41% in 2007, falling to 20.40% in 2008). In a similar fashion, the proportion of students who

passed also increased for those passing with a mark between 5 and 8 (46.94% in 2006, 37.25% in 2007 and an increase in passes of 63.28% in 2008). However, the highest mark range did not experience an improvement with the new methodology.

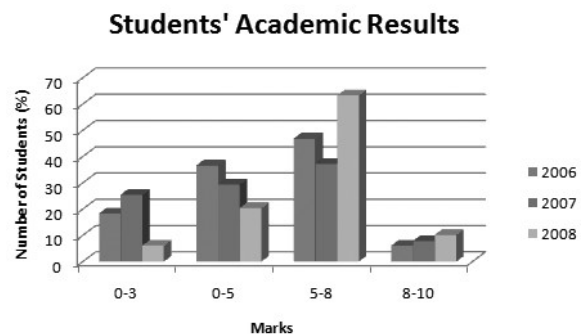


Fig. 4. Students' academic results.

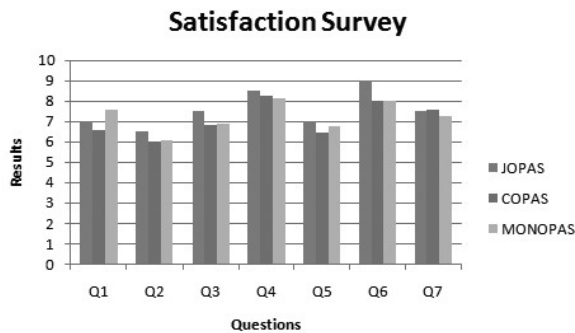


Fig. 5. Satisfaction survey results (average). 9–10. Strongly agree, 7–8. Agree, 5–6. Neutral, 3–4. Disagree, 1–2. Strongly disagree.

4.2 Opinion Survey Results and Student Comments

Student satisfaction was assessed by means of a questionnaire given at the end of each semester. Figure 5 shows the results from the satisfaction survey given to the students. Each item was evaluated on a scale of 1 to 10. In Table 2, each item’s average result can be seen.

The questionnaire was designed to cover the following features: assessment of the tool (three statements), academic performance (two statements), quality of documentation (one statement) and general satisfaction (one statement).

The first three statements, which are related to the tool, received an assessment score of over 6 out of 10. This reflects a positive evaluation, but also a certain degree of difficulty with the initial use of the tool.

Statements 4 and 5 concern the academic value of the tool according to the students. In this case, the evaluation fluctuated between 6.72 and 8.5, with a maximum deviation of 0.74, reflecting a medium–high evaluation of this feature.

The documentation was assessed by statement 6 and obtained an excellent score from the students.

Finally, statement 7 assessed the students’ level of general satisfaction and scored over 7.23, with a

maximum deviation of 0.99. This can be interpreted as a success for this new methodology.

5. CONCLUSIONS

The conclusions reached in this work as regards both fulfilment of the set aims and the results obtained are included in this section. Finally, the challenges and future work to be carried out are summarized.

First of all, the planning of a new signal processing subject in engineering degrees in accordance with the Bologna agreement requirements has been successfully completed. As has been described in this paper, the number of hours that the student should devote to both in-class and out-of-class work has been taken into close account. Here, activities such as attending practical classes, classroom expositions, solving problems, taking a theoretical exam, taking a practical exam, reading and studying the topic, exercises or work time for experiments are all included.

The final objective set by the authors was that of facilitating the development of projects and graphics environments in various programming languages (Java, C++ and C#) with straightforward access to mathematical functions through the Octave/Matlab tool. For this purpose, three independent applications implementing the ‘bridges’ between each of the previously mentioned high-level languages were developed within the MultiPAS tool. The bridge application for connecting Java with Octave/MATlab is called ‘jopas’. A bridge called ‘copas’ was implemented for C++, and finally, there is an application known as ‘monoPAS’ for C#. All of these were integrated into the MultiPAS tool that is given to the engineering student collective. Every student can therefore continue using the high-level language with which they are accustomed to programming graphics environments and can use this technology to integrate signal processing algorithms in

Table 2. Opinion survey results

QUESTION ASKED (of a group of 35 students)	joPAS		coPAS		monoPAS	
	Avg	Std	Avg	Std	Avg	Std
Q1—The gateways cover all of the operations of signal processing.	7.01	0.34	6.55	0.89	7.57	0.74
Q2—The degree of difficulty/ time needed to master the gateways is low.	6.53	0.78	6.00	0.89	6.04	0.79
Q3—The degree of difficulty/ time required to develop digital signal processing systems in [Java, C or .NET] is low.	7.50	0.23	6.80	0.69	6.89	0.98
Q4—The gateway design allows for a thorough study of the subject content.	8.50	0.76	8.23	0.34	8.10	0.35
Q5—The gateways are more motivating/easier to use than the master class and exam method.	7.00	0.74	6.45	0.62	6.72	0.66
Q6—The documentation on the gateway is clear.	9.00	0.50	8.00	0.61	8.00	0.35
Q7—General satisfaction.	7.50	0.99	7.57	0.67	7.23	0.59

Octave/Matlab into those they will access through MultiPAS.

The experience described in this paper, the introduction of PBL methodology to the subjects of signal processing in the areas of telecommunications and electronic engineering has proved to enhance the students' level of learning and performance. In particular, the newly proposed method was efficient in improving the performance of those students having greater difficulty in this area.

The number of students failing the subject decreased considerably: the total percentage of students passing the subject rose by 20.42%, as compared with 2006, and 28.38%, as compared with 2007. The authors consider these results to be very positive, especially given that they are not only positive from an academic point of view, but also that the feedback received from the survey demonstrates medium-high general satisfaction with the method on the part of the students. In addition, the professors involved in teaching the subject—using both the traditional method previously and the new method in 2008—expressed great satisfaction with the results and a willingness to apply the same methodology the following year.

Having obtained these successful results, MultiPAS has been published at Sourceforge.net for use and assessment by the scientific community.

The challenges identified for the next phase can be broken down into two:

1. Development of new bridges that are integrable into MultiPAS with a view to increasing

the number of high-level languages accessing MultiPAS. It would be interesting to include php [28], for example, which is a web language, so that web demo applications can be performed using calculus in Octave.

2. Development of a new tool that implements bridges for these high-level languages with other digital signal processing or mathematical tools apart from Octave/Matlab. An example of such possible digital processing tools would be Scilab [29] and Yacas [30].

Finally, a future line of work we have planned to deal with is corporate progress assessment of students. When engineering graduates enter the labour market, we will be able to carry out surveys on companies in our area, so as to consult them on the competences and skills learnt by students following the new methodology, in comparison with the traditional methods. As a first step, approval has been received from GAIA [27]—Association of Electronics and Telecommunications Companies in the Basque Country—to carry out satisfaction surveys in its associated companies.

*Acknowledgements*—this work was partially supported by the Basque Country Department of Education, Universities and Research. The authors of this paper would like to thank the University of Deusto for the support it gives to this kind of initiative through the application of pedagogical innovation projects in 2007 and 2008. We would also like to express our gratitude to Sourceforge (www.sourceforge.net) for allowing us to house the APIs in their servers, thus enabling anyone wishing to download them to do so from the Internet.

## REFERENCES

1. S. Bergan and A. Rauhvargers (eds), *Recognition in the Bologna Process: Policy Development and the Road to Good Practice*, Council of Europe Higher Education, series 4, 2006.
2. E. Erdil and A. Bilsel, Curriculum design to revitalise electrical engineering education at eastern Mediterranean university, *Int. J. Electr. Eng. Educ.*, **42**(3), 2005, pp. 234–246.
3. *ECTS User's Guide, European Credit Transfer and Accumulation System and the Diploma Supplement*, Directorate General for Education and Culture, Brussels, Belgium, 2005.
4. John G. Proakis and Dimitris K. Manolakis, *Digital Signal Processing*. 4th edn, Prentice Hall, 2006.
5. Alan V. Oppenheim, Ronald W. Schaffer and John R. Buck, *Discrete-Time Signal Processing*, 2nd edn, Prentice Hall, 1999.
6. L. Sales, J. Berral, I. Serrano and P. Martínez, An assessment of the ECTS in software engineering: a teaching experience, *IEEE Transactions on Education*, **52**(1), 2009.
7. Kurt Hornik, Friedrich Leisch and Achim Zeileis, Ten years of Octave recent developments and plans for the future, *Proc. DSC 2003*, Vienna, Austria, 2003.
8. B. Stroustrup, *The C++ Programming Language*, Addison Wesley, Special Edition, 2002.
9. B. Powell and R. Weeks, *C# and the .NET framework: the C++ perspective*, Sams, Indianapolis, Indiana, 2002.
10. Ken Arnold and James Gosling, *The Java Programming Language* The Java Series, Addison-Wesley, Reading, MA, 1996.
11. G. M. Huang and H. Zhang, A new education MatLAB software for teaching power flow analysis that involves the slack bus concept and loss allocation issues, *Power Engineering Society Winter Meeting, 2000*, IEEE, Volume 2, 2000, pp. 1150–1155.
12. C. Levy-Leboyer and J.M. Prieto, *Gestión de las competencias: cómo analizarlas, cómo evaluarlas, cómo desarrollarlas*, Gestión, Barcelona, 2000, 1997.
13. K. M. S. Soyjaudah and M. I. Jahmeerbacus, A New Digital Communications Course Enhanced by PC-Based Design Projects, *Int. J. Eng. Ed.*, **16**(6), 2000, pp. 553–559.
14. Alexandra Branzan Albu and Kaveh Malakuti, Work in Progress—Problem-Based Learning in Digital Signal Processing, *Proc. ASEE/IEEE Frontiers in Education Conference*, San Antonio, 2009.
15. M. L. Ho, A. B. Rad, and P.T. Chan, Project-based learning, *IEEE Control Systems Magazine*, Oct. 2004, pp. 88–91.



16. D. Davenport, Experience using a project-based approach in an introductory programming course, *IEEE Transactions on Education*, **43**(4), 2000, pp. 443–448.
17. Jill K. Nelson, Work in progress: project-based assignments for a graduate-level digital signal processing course, *Proc. ASEE/IEEE Frontiers in Education Conference*, San Diego, 2006.
18. P. Shankar and B. Eisenstein, Project-based instruction in wireless communications at the junior level, *IEEE Transactions on Education*, **43**(3), 2000, pp. 245–249.
19. R. Hsu and W. Liu, Project based learning as a pedagogical tool for embedded system education, *Proceedings of the Third International Conference on Information Technology: Research and Education*, June 2005, pp. 362–366.
20. Madeleine Dahlgren, PBL through the looking glass: comparing applications in computer engineering, psychology and psychotherapy, *Int. J. Eng. Educ. Special Issue* **19**(5), 2003.
21. Bernd Bender and Jörg Longmuss, Knowledge Management in Problem-Based Engineering Design Projects, *Int. J. Eng. Educ. Special Issue*, **19**(5), 2003.
22. Terra Smith and William S. Janna, Reflections on scholarship of integration as a model for problem-based learning in undergraduate engineering education, *Int. J. Eng. Educ. Special Issue*, **19**(5), 2003.
23. J. Sánchez, E. Esquemere, C. Martin, S. Dormido, S. Dormido-Canto, R. D. Canto, R. Pastor and A. Urquia, Easy Java simulations: an open-source tool to develop interactive virtual laboratories using MATLAB/Simulink, *Int. J. Eng. Educ. Special Issue*, **21**(5), 2005.
24. J. Tung, MATLAB and Simulink in engineering education, *Int. J. Eng. Educ. Special Issue*, **21**(4), 2005.
25. J. Z. Zhang, R. Adams and K. Burbank, Theory, practice and systems—a new approach to teaching electronic communications with MATLAB, *Int. J. Eng. Educ. Special Issue*, **21**(4), 2005.
26. D. Harris, A case for project-based design education, *Int. J. Eng. Ed.*, **17**(4), 2001, pp. 367–269.
27. GAIA, [www.gaia.es](http://www.gaia.es)
28. Tobias Ratschiller, *Web Application Development with PHP 4.0*, Pearson Education, S.A, 2000.
29. S. Campbell, J.-P. Chancelier and R. Nikoukhah, *Modeling and Simulation in Scilab/Scicos*, Springer, New York, 2006.
30. Ayal Z. Pinkus and Serge Winitzki, YACAS: a do-it-yourself computer algebra system, *Lecture Notes in Artificial Intelligence*, **2385**, Springer-Verlag, 2002, pp. 332–336.

**Begoña García Zapirain** graduated in Communications Engineering, specialising in Telematics at the Basque Country University in 1994. In 2003, she defended her doctoral thesis in the pathological speech digital processing field. After many years of working for ZIV Company, in 1997 she joined the University of Deusto faculty as a lecturer in signal theory and electronics. She has been the head of the Telecommunication Department at the University of Deusto since 2002. In 2001 she created the Advanced Signal Processing (PAS) research group along with Javier Vicente Sáez from the same university, playing the role of chief researcher.

**Amaia Méndez Zorrilla** obtained her MSc in Communications Engineering from the University of Deusto, Spain, in 2001. Since 2003, she has formed part of the Computer Architecture, Electronic, Automation and Communication Department at the University of Deusto. She became a member of the Advanced Signal Processing (PAS) research group at UD in 2005.

**Ibon Ruiz Oleagordia** graduated in Physics at the Basque Country University in 1999 and obtained his degree in Electronic Engineering at the same university in 2001. He has his doctoral thesis registered. Since 2000, he has worked as a lecturer and has been part of the Computer Architecture, Electronic, Automation and Communication Department at the University of Deusto. In 2002 he became part of the Advanced Signal Processing (PAS) research group at Deusto University.

**Javier Vicente Sáez** graduated in Technical Industrial Engineering, specialising in Electronics in 1999 at the University of Deusto (UD). In 2001 he graduated in telecommunications engineering at the same university. He has his thesis registered in Digital Signal Processor Architectures. He has been a lecturer at UD since 2001 and is part of the Telecommunication Department at UD. He joined the Advanced Signal Processing (PAS) research group at UD in 2001.