Work in Progress - Problem-Based Learning in Digital Signal Processing

Alexandra Branzan Albu and Kaveh Malakuti
Dept. of Electrical and Computer Engineering, University of Victoria (BC), Canada, aalbu@ece.uvic.ca

Abstract – **Learning core concepts in signal processing courses is difficult for undergraduate students in electrical engineering. In our opinion, this difficulty comes from the gap between understanding the mathematical formalism of such concepts and being able to make sense of them in a practical way.**

To bridge this gap, we investigate a hybrid instructional format that combines traditional lectures with a problem-based component. Our initiative aimed at providing a framework for solving multimedia-related DSP problems, where students were encouraged to formulate their own problems.

The paper describes in detail the PBL component that was added to the lecture-centric course and provides an overview of the custom-designed software for system design and analysis. It also discusses the lessons that we learned during the first offering of this new course format and outlines future work directions.

Index Terms – Problem-Based Learning, Digital Signal Processing

1. INTRODUCTION

Learning core concepts in signal processing courses is a challenge for undergraduate students. One possible cause for this is the gap between understanding the mathematical formalism of such concepts and making sense of them in a practical context.

Signal processing is an applied engineering field, although it relies heavily on mathematical concepts. Therefore, we investigate whether Problem-Based Learning (PBL) in combination with a lecture-based teaching style is able to enhance the learning outcomes of an introductory core course in digital signal processing. This paper reports on a pilot pedagogical project that combined for the first time problem-based learning with a lecture-based instructional style in Digital Signal Processing I.

It is important to mention that our implementation of problem-based learning is different from the experiential learning, or the learning-by-doing approach that is usually implemented in engineering classes via laboratory assignments. The problems to be solved in such assignments are specified in closed form by the course instructor, and they follow closely the theoretical content of the lectures. Our initiative aimed at providing a framework for solving multimedia-related DSP problems, where students were encouraged to formulate their own problems. More details are to be found in section 2.

The remainder of the paper is structured as follows. Section 2 describes the PBL component that was added to the lecture-centric course. Section 3 provides an overview of the custom-designed software for system design and analysis. Section 4 discusses the lessons that we learned during the first offering of this new course format and outlines future work directions

2. PROBLEM-BASED LEARNING APPROACH FOR DSP I

Problem-based learning is becoming increasingly popular in engineering. It is typically implemented via engineering design projects, as shown in [1-2]. Projects represent the hands-on component of engineering courses, where students apply theoretical knowledge in the context of a specific application. However, formulating an appropriate project for an introductory DSP course presents challenges on how to avoid coursework overload. Since introductory DSP courses are typically very dense from a theoretical viewpoint, the PBL component needs to be carefully crafted in order to complement well the theoretical course content without overloading the student.

Our hypothesis is that while observing how simple DSP components (e.g. 1st or 2nd order filters, delays and up/down samplers) work on familiar signals students will grasp more easily fundamental concepts such as superposition, convolution, frequency response, sampling etc. The programming environment (see Section 3) enabled the direct perception of how filters change the input signals.

The projects were formulated so that the students had to achieve a well-defined goal. This goal was reachable via multiple paths, i.e. by choosing among several multi-step signal processing approaches. Two example problems were proposed by the instructor. Students were encouraged to specify their own problem within the area of audio/image processing.

Problem A) Enhancement of speech signal

Speech signals are recorded using a microphone with unshielded cable. As a result, the signals are contaminated with 60Hz interference. Also, due to the fact that the microphone's frequency response is not flat and that higher pitches are suppressed, the recorded voice sounds humming, with a background buzz.

Students were to design a DSP system to enhance the signal-to-noise ratio of their speech signal and to modify its spectrum in such a way that it sounds as natural as possible.

In the planning stage, students needed to select their signal processing blocks and to justify their selection. The implementation stage involves the interconnection of the blocks and the parametric adjustment of each block.

Problem B) Enhancement of print-out quality of facial images

Facial images are captured using a low quality webcam. The images feature Gaussian noise and are over-exposed, due to excessive lighting. The quality of the images needs to be optimized for printing purposes.

The planning stage needs to identify the processes to be performed (such as de-noising, contour sharpening, histogram equalization, removal of specular regions etc.). The implementation stage involves interconnecting the processes above, as well as tuning their parameters.

All problems (either instructor- or student-formulated) were to be solved using EasyDSP, a custom-designed software described in the next section.

The project deliverables consisted of:

- A written report describing the problem, its solution, and the design methodology.
- The complete, functional system saved in EasyDSP format so that it could be tested for marking purposes.
- All signals used in system design and testing.

The project counted 5% of the total course mark, with the possibility of gaining up to 5% extra points for high-quality projects dealing with original problems.

3. The EasyDsp environment

The main goal of the EasyDSP environment is to enable a fast and easy prototyping/design of DSP systems for image and audio processing. The paradigm that we propose is direct visual manipulation of discrete signals and systems. It is similar to the WYSIWYG (What You See Is What You Get) paradigm, a very popular choice for text editors. The graphical interface of EasyDSP allowed students to manipulate the parameters of each incorporated block for digital signal processing and thus to gain understanding of how it affects the input signals.

The development of a system in EasyDSP consists of the following steps: a) Loading or creating the input signals; b) Creating filters; c) Interconnecting sources, filters and arithmetic blocks; d) Placing probes at desired points in the system for signal investigation; e) System simulation; Viewing and saving the outputs.

Filter design has been implemented via dropdown menus and sliders for choosing filter type and parameters. Once the filter is designed, it can be visualized using a pop-up window. Students can investigate magnitude, phase, impulse and unit step responses of their filter as well as zero-pole, group delay and phase delay plots. These filter design options make EasyDSP more intuitive than Matlab's FDATOOL, since they enable novice users to locate, manipulate, and visualize their filters faster, without getting lost in advanced options.

EasyDSP also provides the option to probe the signal at any desired location in the system by using pop-up windows or by saving it to a file. Probes are considered a signal sink and can be placed at the output of any block on the system. The simultaneous time (or space)-domain and frequency-domain visualization is useful for understanding the correlation between time- (space-) and frequency domain representations, as well as the trade-off between different constraints imposed on filters in either time, space or frequency.

Lessons Learned and Future Work

The first offering of this new hybrid course format has been an intense learning experience for the instructional team. We present here the most important lessons that we learned during our pilot project.

The challenging aspect of adopting the PBL style for large classes and introductory core courses has been confirmed by our experience. Our class had over 70 students enrolled, and the instructional resources for the PBL component were limited to one teaching assistant and the course instructor. Along with direct meetings, on-line live help was made available to students for their projects. However, this feature was used by less than 50% of the enrolled students. We attribute this low popularity of on-line live help to the fact that this feature is not used for other ECE courses at UVic, and students need time to adopt it.

Student-formulated problems were found in approximately 15% of the total number of handed-in projects. Interesting applications included image segmentation via edge detection, creating audio effects via digital filtering, and segmentation of piano sounds based on harmony analysis.

The low percentage of student-formulated problems is not surprising, since DSP I students had little exposure to creative problem solving assignments prior to taking our course. Also, some student comments expressed the desire to receive a more structured project assignment formulated in closed-form. Efforts will be made to reverse this preference about passive, directed learning in order to stimulate students' creativity, accountability and involvement in project work. Such efforts will include more in-class project-related discussions and brainstorming sessions.

The EasyDSP software needs improvement in terms of stability, flexibility, and efficiency for the implementation of user-formulated filters. To help students become familiar with EasyDSP, the improved software will also be used for in-class demonstrations and discussions that will complement the lecture content. This will provide a common element between the lecture-based course content and its PBL component.

References

[1] M.L. Ho, A.B. Rad, and P.T. Chan, "Project-based learning", *IEEE Control Systems Magazine*, Oct. 2004, pp. 88-91.

[2] D. Davenport, "Experience using a project-based approach in an introductory programming course", *IEEE Transactions on Education*, vol. 43(4), 2000, pp. 443-448.