The teaching synergy of digital music and SP

Roxana Saint-Nom and Daniel Jacoby
Department of Electrical Engineer - BUENOS AIRES INSTITUTE of TECHNOLOGY (ITBA)
Av. Madero 399, Buenos Aires – 1106- Argentina
saintnom@itba.edu.ar jacoby@itba.edu.ar

ABSTRACT

Teaching signal processing to senior students can be an enjoyable task, provided that they already have good background in MATLAB programming and filter theory.

We describe in this paper a laboratory assignment where students have to synthesize music in different ways: additive, FM, through physical modeling and using wavetables.

In each case we emphasize SP concepts involved and we show good examples of the results achieved.

With small but certain improvements over the last few years, we keep increasing the number of future professionals that will work in DSP.

1. INTRODUCTION

Signal Processing is a box full of surprises. Not only for what it is capable of, but mostly for the way people become fond of it. Teaching SP is not an exception. Simple concepts dressed with some good ideas can develop intense commitment to a subject. We found such experience introducing digital music synthesis in a class of Signal Processing.

We are professors at ITBA in a course of Signals and Systems for senior undergraduate students [1].

We think that laboratory hands-on experiments are the best way to learn, and certainly more fun. Among several subjects we get involved, computer music has a special effect. Most students go far beyond what they are asked because of their high motivation.

Our first steps in digital music begun five years ago when we wrote a simple MATLAB [2] program to implement a plucked string note, based on a proposed implementation in [3]. It was utterly successful in that semester, as they wanted more than just a note. This request came of quite encouraging; therefore we decided to strengthen our background, reading the known book from McClellan et al., DSP First [4]. The way the authors introduced computer music to students, as a teaching tool, showed us that we were in the right path.

When the next year came, we were prepared to broaden our synthesized music experiments, as we showed in [1]. But we didn't stop there, because the movement music created kept on growing. The books from Richard Moore [5] and Curtis Road [6] completed our intuitive grasp of the subject.

The next items of this paper are organized as follows:

The second part will describe the experiments the students must accomplish, pointing out the goal of each one. The third will show examples of student's assignments, chosen among the more interesting results.

The fourth item we analyze the advantages of such experiments, giving results obtained from our acquaintance through the years. Finally we expose our conclusions.

2. CREATING THE CHALLENGE

Before a brief description of the assignments, it is important to clear up that the theoretical concepts required to a complete comprehension of the exercises have been developed and evaluated.

The students are also familiarized with MATLAB, being very skilful programmers, even with Graphical User Interfaces.

2.1. Spectrograms

Full understanding of processed music is not complete without the study of spectrograms. We also think that spectrograms are a good tool to practice DFT concepts and to introduce filter banks. We profit from the MATLAB function `specgram`, which the students must master. To reach to DFT and time limitations, they play the overlapping parameter and different windows.

Although results are quite obvious to audio processing professionals, they are difficult to students. Further use of spectrograms in the next exercises helps them reinforce the concept.

2.2. Additive Synthesis

The equal tempered scale used in keyboard instruments forms the basis for modern western music. It is also the basis of our experiment, grounded on [4].

Students have to reproduce a single scale in different ways. Note envelopes are easily discovered when they listen to it for the first time. From a pure windowed sinusoid to a harmonic signal modulated with an organ-like envelope, a set of odd sounds can be created.

In addition, they are encouraged to play with input signals like square waves, saw-tooth or sinusoids with exponentially decaying harmonics; they also try to emulate actual instrument envelopes that become humble guitars and pianos. In order to complete the picture, we have required from them the processing of the first pentagrams of a well known classical music composition. In most cases, the performance achieved was wholly acceptable.

2.3. FM Synthesis

This exercise is mostly centered on a quick review of FM modulation. Its interesting spectrum properties are thoroughly checked with spectrograms, as well as characteristic sounds of wind instruments and bells. In this exercise, our starting point was [4] again, but we developed the assignment a little further.

In order to obtain nice sounds, somewhat near a wind instrument, the choice of envelope and frequency parameters is crucial. At this point students are asked to research through literature and the Internet, with the purpose of enhancing their synthesized sounds.

They have found very hard to cope with clarinets and saxophones; on the other hand, oboes turned out to be easier to grasp.

2.4. Physical Model Synthesis

This kind of instrument synthesis is a complex task. It is not only difficult to obtain a good sound quality but even harder to understand the model. That is why they are out of the scope of our course. Fortunately and thanks to Karplus and Strong [7], there is and extremely simple model of a plucked string. Its block diagram can be seen in Figure 1. We have been using it for years, therefore incorporating some upgrades that promoted interesting outcomes.

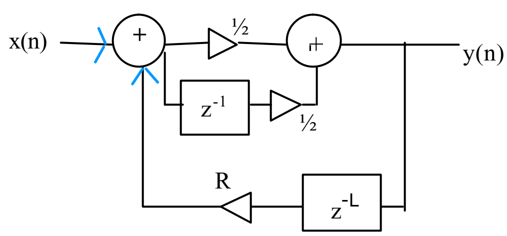

Figure 1. A plucked string model

The first goal of physical modeling was to combine digital filter theory with production of sounds. However, it turned out to be much more than that. Among the subjects involved are:

- Comb filtering
- White noise processing
- Loop stability
- Phase equalization
- Low-pass filtering, to emulate in-instrument absorption of sound waves
- Excellent example to test Simulink's capabilities

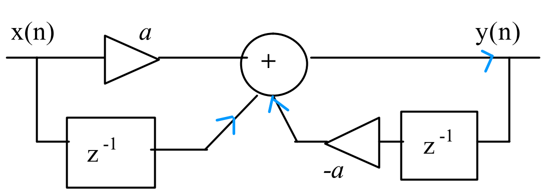

Figure 2. An all-pass filter

The variety of issues addressed comes from other sources such as the introduction of an all-pass filter in the loop, as the one in Figure 2, with the idea of obtaining exact note pitch, regardless its ratio to the sampling frequency. The parameter a has the constraint: $0<a<1$ [8]

We also tried a new scheme proposed in [9], that gave interesting results in many cases, but that proved to have some limitations. Figure 3 displays it, in order to compare its differences to the preceding figures. The most interesting feature of this procedure is its model complexity, creating a set of limitations that requires student involvement.

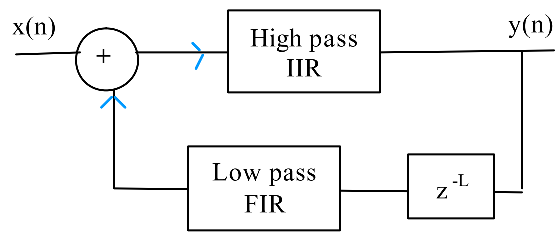

Figure 3. Sullivan's model

2.5. Wavetable Music Production

A short introduction to music synthesis wouldn't be complete if we omit wavetables.

When the subject was not part of these laboratory experiments, we were always asked about it. Mostly due to computer music and sound-blaster card capabilities, we were forced to implement an exercise.

We explored the internet to find out music data and simple code to process those files. Music notes from different instrument were modulated by envelopes created by the students

When the scales chosen had good resolution, the performance of synthesized music was outstanding.

Although at this point, no deep SP concepts were added, the sense of accomplishment the students got after the experience, made the whole exercise worth.

3. Solved Examples

In this item we will expose some student work, to illustrate the points detailed before. We intend, by no means, to explain the contents or the way they took to reach the objectives. The interested reader can request sound examples or any didactic orientation regarding the exercises from the authors.

Spectrogram training is first performed using an equal tempered scale. After some efforts, they are prepared to implement a melody, as the example displayed in Figure 4. In this case an organ-like envelope was employed. The right choice of spectrogram parameters as well as a convenient envelops to emulate the instrument, are the main goals of the exercise. Although the audible results are not similar to a true instrument, they play pleasant sounds, and develop in students the idea of going further.

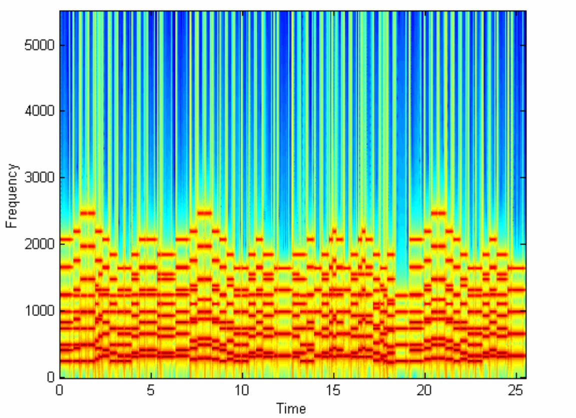

Figure 4. Spectrogram of a melody

Beginning with a FM modulated signal and a set a parameters carefully chosen to create each instrument, students have programmed a MATLAB script which has a GUI showed in Figure 5. The results obtained, along with the wanted sound, include spectrograms and time characteristics, which allow them to associate sounds with displayed signals.

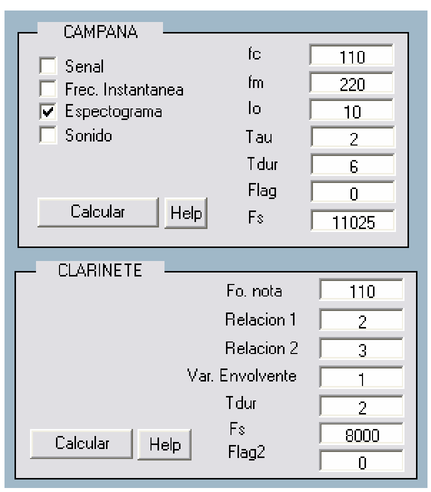

Figure 5. GUI Interface of FM Synthesis

Physical modeling of plucked strings has many faces, depending of the algorithms employed. The GIU of Figure 6 features a guitar chord performed with a Sullivan algorithm The advantage of such script depends on its flexibility. In this example, the only purpose is to play the synthesized sound, but there are other capabilities not shown there, such as:

- Pole-zero plot of the first note
- Pitch correction
- Spectrogram
- Impulse response
- Previous noise modeling, to ease string plucking.

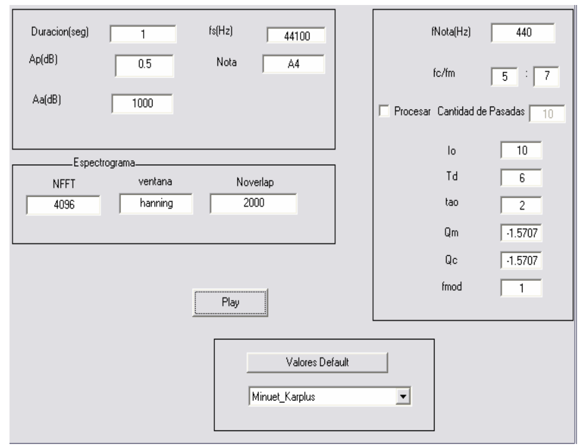

Figure 6. Algorithm flexibility

In case of wavetable synthesis, high resolution data of a complete scale of a chelo was used. The desired sound was achieved performing a resample of the appropriate signal, and its duration had to be chosen according to the instrument's envelop. Figure 7 exhibits this envelope created. It is interesting to point out that the front-end of this MATLAB script allows the performance of a MIDI file with a preloaded instrument data. Provided there are envelop parameters to switch, dramatic changes in the output melody may be produced.

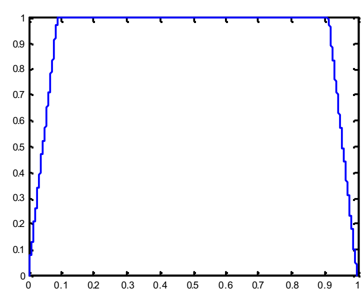

Figure 7. Chelo envelop

The last example is what they called "a music studio". The script behind its GUI is more developed than what was asked, combining capabilities of Simulink and MATLAB.

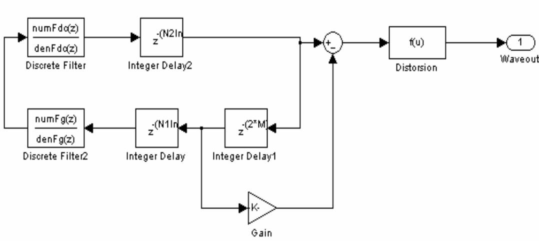

Figure 8. Simulink model of a plucked string

Figure 8 shows the plucked string model used. It is also interesting to point out that most of the previous assignments can be performed by the same program. It is possible, for example, to play a selected double pentagram melody with different instruments: a guitar, an oboe, an organ, a simple bell, or even a self-made sound.

4. OUTCOME

4.1. Qualitative results

The assignment takes about a month to be completed. We split the course in groups of four students. In each of them, tasks are distributed according to personal skills. One becomes the programmer leader, while another accomplishes the theoretical matters. Each group contains a musical "advisor" and a "researcher". Such a role play contributes to build a strong group relationship, always required in their future carrier development.

It is a fact that not all students are fond of the project. But compared to other tasks they performed along the semester, this one has high percentage of acceptance, above 70% of the class. This result, shown in the next figure is obtained from surveys taken at the end of each course.

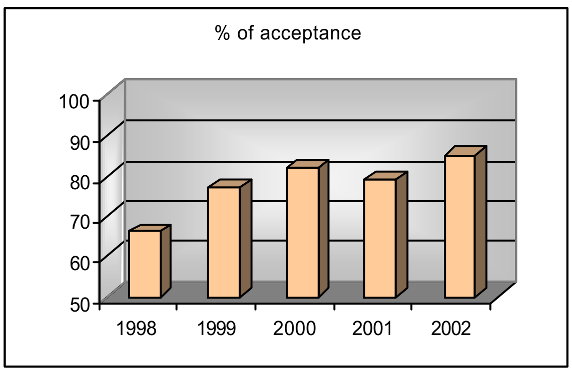

Figure 9. Acceptance rate of the assignment

4.2. Open creativity

When some students get actually involved with the assignment, and it happens very often, we encourage them to investigate further, and to upgrade the MATLAB code.

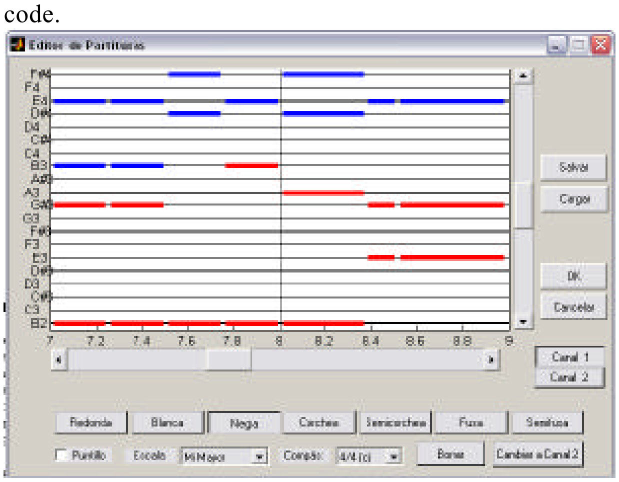

Figure 9. Double pentagram editing

In some cases, they also developed a graphic interface to introduce music notes from pentagrams, taking the performance of programs like the one of Figure 8 even further. They built MATLAB modules that actually read music from MIDI files downloaded from the Internet, with edition ability, as seen in Figure 9.

The experience of correcting these assignments could be called enjoyable.

5. CONCLUSIONS

We described a MATLAB assignment intended to learn, through music synthesis, several signal processing topics.

We showed chosen examples of former student work, where the proactive benefits of the exercises can be easily acknowledged.

A comb filter, a z-plane singularities plot or a spectrogram are important but cold SP concepts.

However when they are blended into synthetic music, they awake student eagerness to learn. We think that promoting that feeling is what our job is about.

ACKNOWLEDGEMENTS

To our students

The following former SP students provided us with the examples:

- García Del Rio, Diego; Lapponi, Diego; Pravisani, Bruno; Pupillo, Esteban; Spina, Alberto
- Balaciano, Pablo, Forconesi, David; Mongini, Juan Pablo; Rozenbaum, Uriel;
- Garcia Sereno, Patricio; Gastiazoro, Francisco; Mac Laughlin, Guillermo; Oliverio, Jorge Luis

REFERENCES

[1] "*Nice Experiences teaching SP in Argentina*", **Jacoby** D., **Saint-Nom** R.- ICASSP 2001 Proceedings, *Vol. V*

[2] **MATLAB** ®:http://www.mathworks.com/products/

[3]"*A Digital Signal Processing Primer*", **K. Steiglitz**, Addison-Wesley, 1996.

[4]"*DSP FIRST – A Multimedia Approach*" **McClellan, Schafer and Yoder** – Prentice Hall 1999.

[5]"*The Computer Music Tutorial*" **Curtis Roads** MIT Press, 1996.

[6]"*Elements of Computer Music*" **F. Richard Moore,** Englewood Cliffs, NJ: Prentice-Hall, 1990.

[7]"*Digital Synthesis of Plucked String and Drum Timbres*" **Karplus, K., and A. Strong**. 1983. Computer Music Journal 7 (2):43-55.

[8]"*Extensions of the Karplus-Strong Plucked-String Algorithm*". **Jaffe, D.A. and J.O. Smith.** 1983. Computer Music Journal 7(2):56-69.

[9]"*Extending the Karplus-Strong Algorithm to Synthesize Electric Guitar Timbres with Distortion and Feedback*". **Sullivan, C.R.** 1990. Computer Music Journal, Vol.14, Nº3.