One-Shot Cost Conscious Digital Signal Processing

T.Aboulnasr
Ottaw-Carleton Institute of Electrical Engineering
University of Ottawa
Dept of Electrical Engineering
Ottawa, Ont. Canada
K1N 6N5
email: aboulnas@trix.genie.uottawa.ca

ABSTRACT

This paper summarizes our approach teaching Digital Signal Processing at The University of Ottawa. We are restricted to one fourth-year all-inclusive course in which we try to cover all the basics of DSP while introducing students to software/ hardware tools and applications. It is our belief that based on a balanced combination of DSP packages, high level programming AND hardware components; we are able to achieve our objectives in the limited time available.

1. INTRODUCTION

This paper presents our experience at the University of Ottawa in teaching Digital Signal Processing at the undergraduate level. The wealth of information in that field is directly in conflict with the fact that in many universities, exposure to discrete-time signal processing is restricted to a one-shot all-inclusive fourth year course. At our university, we do not have the luxury of teaching the basics in earlier years and challenging the students with more advanced topics in the final year [1]. When they arrive at the fourth year, our students are only marginally exposed to discrete-time systems by way of the sampling theorem and basic representation of discrete-time convolution as an introduction to continuous-time convolution. Within the span of approximately 13 weeks (3 hours class/week plus laboratory), they are expected to have mastered time, frequency, z-domain representations, DFT and FFT, spectral analysis, filter design, basic multi-rate processing, A/D and D/A conversion issues as well as actual applications. All this has to be achieved while being familiarized with software/hardware tools to be used to design and implement various systems: a considerable task for both students and professors.

The paper will briefly outline the course offered along with the approach followed in choosing and timing assignments. The question of high Level programming using C versus Digital Signal Processing packages like Matlab [2] as well as simulations versus real time implement-ations will be addressed. At our university, we firmly believe the combination of all those options provides a more balanced understanding of the subject.

2.COURSE OUTLINE

As stated in the introduction, students start this course with the basic knowledge of the sampling theorem and very little exposure tc discrete-time systems. Following the introductory topics of systems properties and their representation in the time and frequency domains; the sampling theorem is revisited to tie the digital processing system with the possible analog input/output interfaces. A/D and D/A converters are discussed in that context and their limitations acknowledged. At this point, oversampling and its use to overcome some of the limitations of the analog anti-aliasing and reconstruction filters is introduced as an application of multirate processing. This turns out to be one of the most exciting parts of

0-7803-2431-5/95 $4.00 © 1995 IEEE

the course for the average student since they can relate to terms like oversampling and bit-stream based on their familiarity with Compact Disc players. It puts a practical face on sampling, Fourier transform and reconstruction theoretical ideas.

Z-Transform is then introduced as a more general and pictorial method of representation. Determination of all system/signal characteristics from the corresponding Z-transform is then presented. This leads up to the subject of restrictions in one domain and their corresponding implications in the other domain. Examples of allpass, linear phase,..etc. systems are constructed. Next, the design of such systems by pole/zero location is illustrated followed by the more formal FIR/IIR design approaches. The DFT and its use for spectral analysis is then studied along with the associated topic of windowing (to which the students had already been exposed in the FIR design).

Throughout the course, applications are pointed out: subbanded filtering and their use in compression, notch filters and their use in anti-jamming devices, channel equalization, digital graphic equalizers, reverberation, echo cancellation ,etc.

3.SOFTWARE : C OR MATLAB

The question of use of Digital Signal Processing software packages partially or exclusively always arises. In this course, Matlab has been used extensively to illustrate ideas and enhance class discussion through repeated examples. It is instrumental in driving ideas home to the students. Once the students got comfortable with it, they usually used it over and above assignment requirements to answer all kinds of individual *'what if'* questions. Throughout the course, students were quite aware that with the availability of a multitude of DSP packages, they may not need to go through the actual design steps for something like Bilinear transformation as an example. Matlab can do that. However, they had to know how the design is performed so as to be able to understand its capabilities and limitations: when to choose it and at what price. DSP *'gut feeling'* and *'common sense'* are emphasized as the unofficial objectives of the course.

However, despite its enormous power in illustrating ideas, we firmly believe that *'exclusive'* use of Matlab (or other similar packages) severely limits the students' perception of issues that may arise when a real time implementation is necessary. Operations as fundemental as convolution are very neatly packaged in Matlab and it is quite surprising to see how many students have trouble programming it in C the first time for processing of blocks of input. C coding in a case like this naturally leads to the details of convolution implementation : overlap and save versus overlap and add etc. Since C programming will be the final objective for most current chips, Matlab is highly misleading in its simplicity. Thus, students are required to code some assignments in C throughout the course as partial requirements for their project (to be addressed later in the paper).

In general, assignments are used to enhance or expand upon ideas expressed in class. Design by pole/zero location is very effective in bringing together all basic concepts: properties of the Fourier transform, conditions for linear phase, allpass etc.. Effect of coefficient quantization on system performance is also shown through simulations. Issues like clustering of poles, closeness to unit circle, etc. can then be easily examined. DFT and effect of windows on the signal spectrum are very clearly visible through Matlab assignments.

4.HARDWARE LABORATORY

It has been argued that by virtue of the simplicity and economic feasibility of software packages like Matlab, we do not need to have a hardware setup. Students can, it is argued, simulate the most complex system, simulate its input and view its output. Why the hardware then? It has been our experience that there is no substitute for the understanding achieved by actually putting in some real time input to the system and hearing or seeing its output. Despite all the concept verifiaction possible using Matlab, students have a hard time visualizing how the difference equation calculation relates to the processing of some speech signal for example. Understanding that aliasing happens and actually hearing it as it happens are two different things.

In this course, the hardware exposure is purposefully split into two disjoint requirements. The first requirement is that of the laboratory experiments where students 'use' the hardware. All the code is provided for the students to be downloaded on the DSP board. As users, they supply the input and change the DSP code by selecting the appropriate laboratory experiment.Since the system parameters can be modified during real time operation, they can study the output for various different combinations of inputs / system parameters.

The laboratory setup is TMS320-C30 based. The setup is fairly simple comprising a C30 EVM board [3] along with the basic scope/generator and preferrably spectrum analyzer. The experiments are designed to go hand-in-hand with the material covered in class and are only done by the students after the subject is introduced. The objectives are: i) to enhance the understanding of the basic ideas, ii) to expose the students to some points only briefly mentioned in class, iii) to highlight cases where practice differs from theory as well as iv) to familiarize the students as 'users' with available software packages and hardware setups. Students are not expected to produce any code for the laboratory just use the ones supplied.

The four lab sessions are divided as follows :

Laboratory session #1:
This is intended as general introduction to the lab setup. No processing is done. The input is simply sampled and then reconstructed so one can observe the effects of the nonideal anti-aliasing/reconstruction filters. Next, the effect of limited accuracy of the A/D converter is simulated through ANDING the A/D binary output with a mask that limits the number of bits as required . The effect is observed on a sinewave where the different quantized levels associated with the simulated number of bits in the A/D output will be very obvious for reasonably low frequencies. The effect of this quantization is then subjectively evaluated by using audio or speech input.

Laboratory session 2:
In this laboratory, students work on three separate experiments :
1) An oscillater. Here, the objective is to understand the effect of difference equation coefficients (pole locations) as well as sampling frequency on the output frequency of the system. Again the quality of the analog output for different frequencies reflects the effect of the quality of the analog reconstruction filter.
2) Basic modulation by a carrier at half the sampling frequency. Again, students can actually hear the change in the frequency while seeing it on the scope/spectrum analyzer.
3) Discrete-time decimation and interpolation. This experiment illustrates the basics of decimation and interpolation. The input signal is sampled as usual. The digital processing consists of simply retaining one out of every N samples and replacing the others first by zeros (discrete time sampling), then by the last retained sample (Zero Order Hold) and finally by a delayed linear interpolation of the last and the next retained samples (First Order Hold). The students can visually see the digital signal on the digital scope on the computer screen and the corresponding analog signal on the scope. Effect of this decimation /simple interpolation is then studied for various input frequencies and decimation factors. Effect of the quality of the interpolater and the reconstruction filter are also considered.

These two laboratory sessions conclude the analysis part of the course.

Laboratory sessions #3, #4 :
In these two sessions, students design various digital filters using DFDP3 [4] and download the automatically generated code of the designed filter on the board to verify real time operation. Practical issues regarding efficiency of FIR versus IIR as well as stability and quantization issues are highlighted in these experiments.

5. Project

In the final part of the course, the students have to submit a project where they design, code in C and implement in real time a specified system. The project is intentionally independent of the laboratory to decouple the "usage" of DSP hardware from its "coding / verification". It is intended to provide the final step combining all the information acquired in the course and building a functioning system from one analog end to the other. This has proven to be the stumbling block for those students with pre-conceived ideas about the horrors of hardware. As such, the project is organized to allow for partial deliverables while insisting on full professional documentation for all steps. The first requirement is the actual C-code generation and associated verification /documentation. The first hardware step is designed to be reasonably achievable by all students. It requires integration of the A/D,D/A sections supplied to the class with a very simple DSP system multiplying the input by a factor. This helps overcome most psychological barriers. Subsequent stages are gradually more challenging: I) A specific system is to be coded to run uninterupted on the board; examples : Graphic equalization using an IIR/FIR filterbank, echo generation, reverberation,.etc. 2) Once this is working, the next is to allow for transfer of parameters from computer to board during real time operation to change the system selected or change the system parameters without having to recompile and redownload the code. By the end of the course, almost all students successfully implement the uninterupted system with varying degrees of sophistication. Though fewer manage the more advanced deliverables, all develop the confidence necessary to attack actual real life DSP problems. In the process of implementing this project, no matter how simple the actual algorithm is, students develop a sense of completion, they actually tie everything together starting from the analog input and following it every step of the way until it arrives at the analog output. This experience is not achievable by exclusive software use. Finally, the course as a whole and the project in particular are very popular and highly appreciated by the students.

Conclusions

In this paper, we presented the outline of a fourth year course covering, in a single shot, all the undergraduate student exposure to Digital Signal Processing at the University of Ottawa, Canada. The issues of software versus hardware were briefly outlined along with those of DSP packages versus C simulations. The hardware laboratory/project components of this course were also outlined. With a combined enrollment of about 80-90 students for the English/French sections of the course, this course has proven to be very popular among students. Despite being repeatedly characterized as having a 'heavier than average' workload with four laboratory sessions, 8-9 assignments and a project, it is a fairly popular course. It has one of the highest enrollments even though it is not required for several specializations.

REFERENCES:

[1] J.P.Allebach et al, 'Digital Signal Processing with Applications: A New and successful Approach to Undergraduate DSP Education', ICASSP 1994.

[2] MATLAB User's Guide, The Mathworks, Inc.

[3] Digital Signal Processing Applications with the TMS320C30 Evaluation Module, Texas Instruments Inc.

[4] DFDP3/plus, Digital Filter Design Package, Atlanta Signal Processors, Inc.