

Teaching Fixed-point Algorithm Development in a Systems Context

Wayne T. Padgett

Rose-Hulman Institute of Technology

Wayne.Padgett@Rose-Hulman.edu

ABSTRACT

A course on fixed-point algorithm design using a single sideband (SSB) communication project has been offered three times. The course emphasizes system cost and performance to simulate industry constraints. Students have found the course motivational, and there is a significant need for the topic in industry. Efforts to assess the effect of the course on conceptual understanding are discussed, as is a plan for an associated text.

1. INTRODUCTION

Fixed-point algorithm design is usually treated only in minimal fashion in most US curricula. Where it is discussed, it is usually only to define the sources of fixed-point error, without providing enough depth to examine tradeoffs or strategies for minimizing fixed-point effects. The result is that most digital signal processing (DSP) engineers learn about fixed-point effects on the job from experienced mentors, and only minimal tutorial or theoretical materials exist to support engineers without an experienced mentor. A course designed to remedy this situation, ECE 497 DSP System Design, was first offered in 2003 at Rose-Hulman, based on the author's experience with fixed-point algorithm design [1].

This paper is intended to summarize the results of offering the course three times, and to describe the course topics, the labs, and the future directions for further course development. This paper is also intended to encourage the development of similar courses

at other institutions so students will have a better understanding of the subtleties of numerical effects and system performance measurement, and better meet the needs of the DSP industry.

2. COURSE DESCRIPTION

The course is called DSP System Design, not Fixed-Point Algorithm Development, because the only way to give students adequate experience with the tradeoffs and performance issues involved is to build the course around a project. Learning to measure, specify, and adjust the system's performance is a critical element of the course, and it drives the students to deepen their understanding of the fixed-point effects. Although many possible projects could serve well, the course has been based on an SSB communication system which takes input speech at an 8 kHz sample rate and then raises the sample rate to 96 kHz (12x) for SSB modulation at 40 kHz (actually four channels are implemented eventually). The receiver removes out of band interference, demodulates the signal, and then reduces the sample rate back to 8 kHz. The basic design is then extended to operate at four channel frequencies with minimal inter-channel interference, and subject to various (conflicting) performance criteria such as speech quality, SNR, and computational complexity.

In the past three offerings, all of the course project work has been done in MATLAB, primarily using the Fixed Point Toolbox functions. All of the fixed-point system development is done as simulation in

MATLAB with no hardware-specific restrictions other than the assumptions of 16 bit registers and 32 bit accumulators. It would be nice to extend the implementation further into a hardware platform, but this material is yet to be developed, and may prove to be too much content for a ten week term.

When the course was originally conceived, the author reviewed many texts [1] and finally used Ifeachor [2] in the first offering because of its emphasis on implementation, but because the students and the instructor use Oppenheim [3] in the graduate DSP course (ECE580), it was usually easier to reference review material in Oppenheim rather than Ifeachor. As a result, Oppenheim has been used in subsequent offerings, which is popular with students since they need only one text for both courses. However, Oppenheim is not an ideal text for the course either, because it is not intended as a text on implementation, and it does not offer support for the system performance aspect of the course.

2.1 Weekly Topics

Rose-Hulman operates on ten week terms. The main topics of the course are shown below for each week of the term. The ordering of the topics is intended to deliver the necessary information “just in time” to allow the students to progress in their implementation of the labs leading up to the SSB system project.

- Introduction and filter structure review
- Binary arithmetic, quantization noise, and overview of fixed-point effects
- Coefficient quantization, signal to noise ratio (SNR), roundoff noise, filtering random processes
- Overflow, scaling, examination of MATLAB’s implementation of scaling for second order sections

- Single sideband modulation, Hilbert transform, single sideband receiver with interference
- Implementation of Hilbert transform (1 filter + delay, 2 filters, infinite impulse response (IIR)), upsampling and downsampling
- Polyphase, staged interpolation
- Sine/cosine generation (table, oscillator), phase noise
- Estimating power spectral density, hardware implementation issues (processor specialization, field programmable gate array (FPGA) vs processor, architecture vs algorithm, cost, power, yield)
- Project time

“Just in time” delivery of the course topics is important to enhance the students’ sense of the practicality of the material. Students tend to be motivated to take this course because they see it as useful in the “real world,” and their positive impression is strengthened by immediately putting each topic to use in lab.

2.2 Labs

The lab topics are also focused on building up the skills and understanding necessary to succeed in doing the project. Because the first lab session normally occurs after just two days of class, and because the students are often unfamiliar with the debugging and object-oriented features in MATLAB, the first lab is intended to bring all the students to a minimum level of competency in using the debugging features, and in manipulating the filter and fixed-point objects used in MATLAB. The next five labs seek to match simulations with actual fixed-point filters and signals with the theoretical material covered in class. The first six labs are done individually, to ensure that all students become competent with the Fixed Point Toolbox. The later labs are done in groups of two.

In each of the theory comparison labs, care must be taken to isolate the effects of a particular source of error from other types of error so that accurate comparisons with the theory can be made. For example, when measuring roundoff noise it is important to use quantized coefficients in the floating point reference system so that the effects of both roundoff noise and coefficient quantization are not combined.

Lab 7 is really a prologue to the project. The students are given a working implementation of an SSB system in floating point MATLAB code, and they must convert to fixed-point, and make measurements of SNR and computational complexity. The actual project involves not just getting the system working, but optimizing according to performance criteria. The system is extended to operate at four different frequency channels, and measurements of inter-channel interference are included. These performance measures require the students to examine various implementations to see which perform best and which minimize computational cost. The critical element of the project is that it exposes students to real tradeoffs and constraints in a way that is significant to them (their grade is effected). The final lab time is used for a wrap up session. The purpose of this session is to help all the groups benefit from the lessons learned by each group. Each team has a chance to share their intuition and see how their results compare in performance and hours logged from a comparison chart.

- Lab 1 – Advanced MATLAB Tutorial
- Lab 2 – Fixed Point Simulation with Custom Tools and MATLAB Toolbox
- Lab 3 – Measuring Fixed Point Errors and Detailed Simulation
- Lab 4 – Quantization Noise, Filtered Noise Power, and Coefficient Quantization

- Lab 5 – Roundoff Noise and Scaling Tradeoff
- Lab 6 – Overflow and Scaling, Examining Worst Case Signals
- Lab 7 – Fixed-point SSB
- Project – Optimized Multichannel SSB System (2 weeks)
- Wrap up Session

3. LESSONS LEARNED

A key element in the success of this course is its obvious practical application. Students really enjoy learning about topics that are hands-on (even though this project is primarily simulated), and that have a clear purpose. The SSB system used in the course could be applied for underwater voice communication, and various cost and implementation issues are emphasized throughout the course. Student enthusiasm has been a hallmark of the course, and it shares both the practical elements and the student enthusiasm with the infrared communication project offered in communication systems [4].

It is critically important to review the material covered in the prerequisite junior level DSP course (ECE380), and to give heavy emphasis to the fundamentals of random processes at the beginning of the course. The material on filter structures in the prerequisite course barely explains direct form 1, direct form 2, and direct form 2 transposed [3], but does not deal with second order sections at all, and the students retain very little of that material since it has no direct application in a floating point system. Likewise, very few students retain useful skills relating to random processes, even for computing noise power. It has been helpful to introduce this topic in a fairly shallow way early to talk about roundoff noise power and signal to noise ratio, and then return after the project has been assigned to talk more about spectral estimation, and the power spectrum of the roundoff noise. This approach gives the

students time to absorb and use the more basic ideas about random processes before going into more detail.

The order of the material is different now than when the course was first offered. The original idea was to begin slowly with information about architecture (which computer engineers have already been exposed to), FPGAs, etc. and then move into the more mathematical fixed-point issues. However, it quickly became evident that implementation material was not necessary for doing the project, and the project needed to be started as early as possible. Now the course begins with a DSP and random processes review and moves immediately into fixed-point material, and the implementation alternatives material is done at the end of the quarter when the project is wrapping up.

The biggest issue that makes this course unique is the way the project forces the students to deal with the interactions between scaling and roundoff noise, and how to isolate the effects of coefficient quantization from roundoff noise. These are not separate theoretical topics, but parameters to be varied to maximize the system's performance. Once they see that system design is not just a series of cookbook problems, but a set of inter-related decisions, they finally appreciate the need to have intuition about fixed-point behavior, because no one can evaluate every possible design choice. In the most recent offering, the wrap up session helped each team gain from the experience of the others, and showed that there were effective alternatives that they had not tried.

4. FUTURE WORK

A number of enhancements to the course are either in development or under consideration. During the most recent offering, a number of students worked on developing material to support FPGA and C55 DSK implementations of the project. The C55 implementation option

would allow students to see more of the development process, including the benefits of optimization techniques like use of processor-specific intrinsics, or optimized assembly routines.

The FPGA implementation option would allow students to optimize the wordlength as well as the scaling and fractional precision. Since hardware description language (HDL) material is not prerequisite to this course, it would be important to use a tool set that allows a gentle introduction to FPGA implementation. Fortunately, the FPGA implementation is possible to do from within National Instruments' LabVIEW, and consideration is being given to doing the entire project in LabVIEW [5]. Of course, if practical, it would be very helpful to look at further optimization techniques for FPGAs as well.

It is not yet clear whether a ten week term is enough time to adequately cover a full hardware implementation of the project, but it does seem clear that it would be impossible to try to cover both FPGA and DSP implementation. If the future of the course goes extremely well, it might be practical to have some groups do DSP implementation and some do FPGA implementation, and then compare performance at the end. This approach seems like an excellent way to expose students to a wide range of implementation options available to them.

In the next course offering an assessment experiment is planned to use the Signals and Systems Concept Inventory (SSCI) [6] to evaluate the relative effects of the graduate level theory course and the fixed-point implementation course on student understanding of fundamental concepts. The SSCI is already being used in the prerequisite DSP course, ECE380, and it is hoped that the experiment will produce interesting comparisons between student learning after

taking two very different types of elective courses, or both. Faculty at other institutions with both theoretical and implementation oriented DSP electives would be welcome to join the experiment to help generalize the data set.

This course has been implemented with students evaluating their own homework (turning in a copy before solutions are distributed). This method has the advantage of moving students to a higher level on Bloom's taxonomy [7], and a new variation is planned wherein students will be assigned some points according to the quality of their self-assessment comments. So far, this self-grading method has been effective both in getting the students to solve the homework problems, and in having them read the solutions, which contain many tutorial comments. The planned variation of grading the comments was motivated by the fact that some students include very insightful comments and demonstrated that much could be learned in evaluating their own work.

No text currently exists which is a good match for the topics and approach in this course. Therefore, a text is planned which will help students take full advantage of the ideas presented above, and also allow others to implement similar courses with a minimum of overhead.

5. CONCLUSIONS

The course has been effective both at motivating students to learn and at pushing them to a higher level of understanding of both system design and digital signal processing. New enhancements and assessment methods are planned to demonstrate the usefulness of the methods applied in the course. The planned course text will solidify the course into a body of material that can be applied at other schools.

6. REFERENCES

- [1] Padgett, W.T., "An Undergraduate Fixed Point DSP Course," Proceedings of the 2nd IEEE Signal Processing Education Workshop, Oct. 2002, Pine Mountain, GA, pp. 302-305.
- [2] Ifeachor, E.C. and Jervis, B.W., *Digital Signal Processing. A Practical Approach*, Addison Wesley Publishing Company, 1995.
- [3] Oppenheim, A.V., Schafer, R.W., and Buck, J.R., *Discrete-Time Signal Processing*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 1999.
- [4] Padgett, W.T.; Black, B.A.; Ferguson, B.A., "Low-Frequency Wireless Communications System-Infrared Laboratory Experiments", IEEE Transactions on Education, v. 49, no. 1, Feb. 2006, pp. 49 – 57.
- [5] Yoder, M.A., Black, B.A. "A Study of Graphical vs. Textual Programming for Teaching DSP", 2006 ASEE National Annual Conference Proceedings, Session 3232, June 2006.
- [6] Wage, K.E.; Buck, J.R.; Wright, C.H.G.; Welch, T.B., "The Signals and Systems Concept Inventory, IEEE Transactions on Education, vol. 48, no. 3, Aug. 2005, pp. 448 – 461.
- [7] Bloom, B.S., *Taxonomy of educational objectives*, Allyn and Bacon, Boston, MA, 1984.