Implementation and assessment of a studio-style laboratory course in real-time digital signal processing

Nathaniel A. Whitmal, III
Department of Electrical and Computer Engineering
Worcester Polytechnic Institute
Worcester, Massachusetts

Abstract

This paper describes the implementation and assessment of EE 3703, a new lab course in real-time digital signal processing (DSP). The goals of the course are (i) to extend DSP-related concepts taught in core courses, (ii) to provide students with design experience that supports the goals of ABET Engineering Criteria 2000, (iii) to prepare students for further work in DSP, particularly in their "Major Qualifying Project (MQP)," a one-year capstone design experience providing credit equivalent to three courses. EE 3703 has two features that distinguish it from previous real-time DSP courses: its use of the interactive "studio" format, and its use of software engineering principles.

Assessment data for the course was derived from student grades, lab reports, identical pre/post-course assessment exams, and MQP reports. The pre/post-course exams focused on topics covered in both the present and prerequisite courses, and were used to gauge the preparation and progress of each student. The data indicate that the course was generally successful in meeting its goals, and also point to areas of improvement for future offerings.

1. Introduction

DSP has, in recent years, become a driving force in the advancement of multimedia and telecommunications technology. In many applications requiring embedded computing (e.g., fax machines, modems, cellular telephones, disk drives), DSP platforms can provide lower cost and higher computational efficiency than general-purpose microprocessors can. Many students have responded to this demand by pursuing additional training in DSP. Munson and Jones [1], for example, note that the percentage of electrical and computer engineering (ECE) students at the University of Illinois choosing DSP for one of their three required elective courses rose from 50% to 80% over a four-year period. For this reason, DSP-oriented labs and senior design courses have become very popular in recent years. A number of these courses focus on programming of DSP chips: typical examples are given in [2,3,4,5,6,7,8].

This paper describes the implementation and assessment of a DSP-based laboratory course that is tailored to the particular needs of students at Worcester Polytechnic Institute (WPI). In 1970, WPI developed a unique outcome-oriented, project-based engineering curriculum, referred to locally as the WPI Plan [9]. The WPI Plan is distinguished by several features:

- Courses are offered in four seven-week terms, identified by the letters A, B, C, and D. Students typically take three courses per term.

- In addition to courses, each student must complete three comprehensive "qualifying" projects, one of which is in the student's major area. Major qualifying projects (MQPs) are typically completed during the senior year by teams of two to four students, who must specify, design, build, and evaluate complex systems. MQP groups present their results to the faculty during "Project Presentation Day," which is held every year in April.

- Outside of basic distribution requirements, there are no required courses at WPI. Instead, WPI students have the freedom to select electives that help to build expertise in their area(s) of interest and prepare them for the MQP.

In the 1999-2000 academic year, WPI's undergraduate sequence in signals and systems consisted of two traditional courses: EE 2311 (Continuous-Time Signals), and EE 2312 (Discrete-Time Signals). At that time, EE 2312 was the department's only DSP-oriented undergraduate course. A survey conducted in the 1999-2000 B-term session of EE 2312 indicated near-unanimity among 45 students, who felt that a more direct link needed to be built between the theoretical material taught in the course and its target applications [10]. A third course, EE 2801 (Embedded Systems Design), gave students experience in programming microprocessors. Unfortunately, it did not teach students the mathematical principles of DSP, or prepare them for working with DSPs that were programmed in a mixture of C and assembly language.

The comments of EE 2312 students are typical of students at many schools [1], who complain that the traditional courses are abstract, and focus almost entirely on underlying mathematical principles (i.e., differential equations and complex analysis). Hence, they see little relation between these traditional courses and their academic and professional interests. This is a noteworthy concern at WPI, since some students misuse the freedom of the WPI plan to bypass fundamental courses in favor of advanced hardware-oriented courses that appear more relevant to their interests.

EE 3703 was developed to address all of the concerns described above. The recommended background for the course is EE 2312, EE 2801, and CS 1005 (Introduction to C Programming). The objectives of EE 3703 were as follows:

I. To reinforce and build upon principles taught in EE 2312, EE 2801, and CS 1005, and provide future students with the incentive to take those three courses.

II. To prepare students for further work in DSP, particularly in preparation of DSP-based MQPs,

III. To provide further exercise of the abilities to (i) apply knowledge of math, science and engineering, (ii) conduct experiments by measuring and analyzing data, (iii) design a system to meet desired needs, (iv) identify, formulate and solve engineering problems, (v) communicate effectively, and (vi) use techniques, skills, and modern engineering tools necessary for practice of engineering, as described in the ABET 2000 criteria.

The course featured the following three innovations:

1. The course employed the "studio" format pioneered by RPI [11]. Studio courses integrate lecture and group lab work, thus providing students with individual attention and hands-on experience. This approach differs greatly from the more traditional approaches used in [3,

4,5,6,7,8], with separate lecture and lab sessions. A variant of the studio approach was used successfully in [2], where EE 2312 material and real-time DSP material were covered in eleven weeks. The shorter terms at WPI necessitate a slightly different approach.

2. Each student had access to his or her own DSP hardware and software. The DSP hardware platform of choice was the Texas Instruments TMS320C31 DSP starter kit (DSK), which retails for $99 and can be operated from the parallel port of any IBM-compatible PC. The TMS series has been the focus of the labs cited above and of two popular textbooks [12, 13], the latter of which has recently been adapted for use with the C31 system [14]. In contrast to the work cited above, the cost and portability of the DSKs allowed students who purchased them to perform the lab experiments on both their home computer and WPI's lab computers.

3. EE 3703 students are taught to view DSP algorithm design as an application that benefits from use of software engineering principles. Towards this end, students are required to use the industry-standard real-time structured analysis (SA/RT) methods of Hatley and Pirbhai [15] to prepare and submit a preliminary system specification for each assignment. These specifications are subsequently used as both guides for the development of the students' software and documentation for the students' lab reports. The Hatley-Pirbhai approach was chosen over more recent object-oriented methodologies (e.g., UML) for its simplicity and its successful track record in industrial practice.

2. Implementation

EE 3703 was offered as an experimental course twice during the 2000-2001 academic year: once in B-term to16 students, and once in D-term to 15 students. During each term, the class met three times per week in two-hour sessions consisting of both lecture and lab modules. The lecture modules, which varied in length between 20 and 60 minutes, were each supplemented by complementary group exercises in Matlab and/or TMS320C3x programming. An abbreviated syllabus for the course is given below in Table 1.

At the end of the course, students were expected to be able to:

- Use structured analysis methods to design algorithms for real-time DSP applications,
- Use Matlab to simulate behavior and predict performance of DSP algorithms,
- Use DSP hardware to sample continuous-time signals, process them as digital signals, and resynthesize the signals in the continuous-time domain,
- Design and implement FIR and IIR filters that meet given frequency-domain specifications,
- Design and implement conventional radix-2 FFT algorithms,
- Use the FFT for spectrum analysis and fast convolution.

Lab assignments were open-ended design problems, specified to meet the learning outcomes specified above. The first assignment required students to specify a simple algorithm (e.g., Levinson's recursion), code it in both Matlab and C, and test it with simulated data. The second assignment extended the first by requiring students to translate their C code to C31 assembly. The third assignment was a real-time processing task (e.g., swept-sine system identification) that required students to process signals acquired through the DSK's A/D converter in accordance with given timing constraints. The timing constraints required the students to prepare SA/RT control specifications that governed the programming of the C31's interrupt and timer facilities. The fourth assignment used a variant of the algorithm from the first assignment (e.g., DPCM coding and decoding) in a real-time application that used both IIR and FIR filtering. The fifth assignment required students to implement FFT-based convolution and compare its computational complexity to that of standard linear convolution.

Students (working in groups of two or three) were directed to begin each assignment by carrying out a preliminary structured analysis of the problem. The students' analyses resulted in data flow diagrams and requirement dictionaries that described their proposed software designs. Once their documented analyses were handed in, students modeled their systems in Matlab to verify the validity of their approaches, and, after receiving good results, implemented their designs on the boards in a mixture of C and assembly code. This approach (which extends the Matlab-based approach used in [5]) is consistent with good engineering practice, and helped students to eliminate coding errors. Final reports consisted of the final structured analysis (with revisions noted), the Matlab simulations, the C / assembly code, and any evaluation data requested in the lab handout. Grades for each assignment were based on both the report and the students' demonstrations of their code.

3. Assessment

Assessment data for the course is derived from several sources, including lab reports, identical pre/post-course assessment exams, MQP reports, and course evaluations.

3.1. Analysis of student performance

A brief overview of D-term student performance is given below in Table 2, which shows average lab and exam grades for all students passing the course. Final grades were derived from a weighted average of lab grades (60% of final grade) and the best four of six quizzes (40% of final grade). Average lab grades are 90.68 of a possible 100, indicating a high level of performance. The largest variation is seen in the scores of Student #1, who earned an incomplete for failing to turn in three of the labs in a timely fashion. Larger overall variations are observed in quiz grades, which provide an indicator of individual performance. The correlation coefficient between quiz grades and final grades is 0.69 when Student #1's grades are included, and 0.90 when his/her grades are excluded. In the latter case, one can predict final grades with reasonable accuracy using the relation

$$\text{final grade} = 59.957 + 3.6341*(\text{average quiz grade}).$$

The relationship between quiz and final grades is shown below in Figure 1.

Identical closed-book pre-tests and post-tests were also given to the students on the first and last days of class. The tests contained one question on properties of first-order IIR systems, and one on C/C++-based implementations of convolution. These questions (shown in Table 3) were selected for two reasons:

1. They rely on fundamental principles taught in the recommended background courses, and provide a measure of preparation for EE 3703,
2. They are given extensive coverage in EE 3703, and provide a reasonable measure of comprehension of EE 3703 material.

Individual and average scores for each pre/post-test question are given below in Table 4. Pre-test scores were uniformly low; post-test scores indicate modest improvement. The low scores on the two tests indicate that students found the questions difficult, even though they were based on prerequisite material. The lone exception was Student #9, who achieved class-high scores of 85/100 (40 on question 1, 45 on question 2) on both the pre-test and the post-test. It is evident that this student's scores on the test were unaffected by his/her enrollment in EE 3703. In contrast, several other students failed to earned partial credit for their answers: the numbers of these students are given in Table IV. The data in Table IV indicate that, as expected, students were better able to answer pre/post-test questions at the end of the course.

Tables 4 and 5 indicate that students had particular difficulty with Question 1 material, which is taught in EE 2312. Average scores on the pre-test and post-test are 3.71 and 11.14 when Student #9's scores are included, and 0.92 and 8.92 when Student #9's scores are excluded. A standard analysis of variance of pre/post-test scores for the latter case indicates that the before / after difference is very significant ($F(1,26) = 10.83$; $p = 0.2\%$).

Differences between Question 2 pre/post-test average scores and final pre/post-test average scores are also favorable, though not statistically significant. For Question 2, the data in Table 4 indicate that students 1, 2, 3, 8, and 11 displayed dramatic improvements on the post-test scores, while the other students earned equal or slightly lower scores on the post-test. One plausible explanation lies in the grading of Question 2, in which both the syntax and structure of the student's algorithms were evaluated. Since some of the students were more familiar with C++ than C (the language supported by the processor), they wrote their Question 2 pre-test answers in C++. As a result, the errors in their pre-test answers were largely conceptual errors. As the course progressed, they became familiar enough with C to use it on the post-test, but perhaps not familiar enough to avoid making some basic syntax errors. As a result, their post-test scores suffered from both syntax and conceptual errors, resulting in slightly lower scores than achieved on the pre-test. It should be noted that, since the tests were closed-book, students were unable to copy from examples, or use examples to check their syntax and correct any errors.

Performance on the pre/post-tests can be further understood by considering student preparation for the class. A listing of each student's grades in prerequisite classes is shown below in Table 6. The data for EE 2312 indicate that eight of the students had not taken EE 2312 before enrolling in EE 3703, and were seeing the material from that course for the first time. Of the remaining students, only two earned A grades in EE 2312, while three earned B grades, and two earned C grades. Preparation for the second question was considerably better. All but one of the students had received credit for CS 1005 and/or CS 2005 (the follow-up course), with nearly all grades being B or better. Overall, Table 6 indicates that, in several cases, students were able to do well in the class despite having neglected to take one or more of the prerequisites. Nevertheless, these students were able to learn the prerequisite material on an as-needed basis, well enough to do the labs and perform well on the quizzes.

3.2. Use of DSP in MQPs

Students have made good use of EE 3703 material in their MQPs. Of the 16 students enrolled in B-term, eight went on to present MQPs in April; of these, six of them worked on project teams that employed principles from the course. Two of these projects were nominated for the ECE department's MQP competition, with third prize going to a team of students that interfaced the C3x DSK to a Palm Pilot in a cryptography application.

Of the 15 students enrolled in D-term, one student went on to present a DSP-based MQP in April. Results from a questionnaire given on the first day of class indicated that six other D-term students expected to use their background from the course to prepare DSP-based MQPs for presentation in April of 2002.

3.3. Course Evaluations

Course evaluations at WPI consist of forms with statements describing the instructor, text, and facilities in positive terms (e.g., "The instructor was well prepared to teach each class."). For each statement, students are asked to provide one of five answers: SA (strongly agree), A (agree), D (disagree), SD (strongly disagree), or N (not applicable). Evaluations for each course are characterized by the percentage of A and SA answers recorded. Mean A+SA scores for undergraduate courses at WPI are typically around 90% [16].

Student response to EE 3703 has been excellent. Evaluations for the course's initial offering in B-term were good (88% A+SA rating), with general consensus that the course should place more focus on basic theory, and less focus on the particular attributes of the processor. The syllabus was modified to address their concerns. The D-term offering received very high evaluations (96% A+SA rating) that placed it in the top quartile of all D-term undergraduate courses. Student comments indicated that the new syllabus struck an appropriate balance between hardware, software, and theoretical topics.

4. Present status

In the fall of 2001, the WPI faculty accepted EE 3703 as a permanent (non-experimental) course. Prior to offering the course in B-term, the author visited the A-term section of EE 2312 to speak to students about (a) enrollment in EE 3703 (b) the relevance of EE 2312 material for study of DSP. The course was subsequently offered in B-term of 2001 to 28 students, all of whom had taken EE 2312 in previous terms. Evaluation data for the most recent offering will be presented in a future paper.

5. Summary

EE 3703, an experimental course in real-time DSP, was offered in B-term of 2000 and D-term of 2001. EE 3703 used the "studio" format to provide students with both the theoretical knowledge and the hands-on experience needed to program DSP chips for real-world applications. EE 3703 has been successful in achieving its goals of (a) building on fundamental background in signals & systems and embedded system design, (b) preparing students for DSP-based MQPs, (c) teaching students new design skills and methods that are consistent with ABET 2001 goals.

References

[1] D.C. Munson, Jr., and D.L. Jones. Analog signal processing first. *IEEE Signal Processing Magazine*, September 1999, pp. 24-29.

[2] V.K. Ingle and J.G. Proakis. A digital signal processing course based on lecture / laboratory integration. *Proc. ICASSP*, April 1992, pp. 105-108.

[3] R. Chassaing, W. Anakwa, and A. Richardson. Real-time digital signal processing in education. *Proc. ICASSP*, April 1993, pp. 28-31.

[4] K.H. Chiang, B.L. Evans, W.T. Huang, *et al*. Real-time DSP for sophomores. *Proc. ICASSP*, April 1996, pp. 1097-1100.

[5] S.M. Kuo and G.D. Miller. An innovative course emphasizing real-time digital signal processing applications. *IEEE Trans. on Education*, vol. 39, no. 2, pp. 109-113, 1996.

[6] M.D. Zolotowski, J.P. Allebach, and C.A. Bouman. Digital signal processing with applications: a new and successful approach to undergraduate DSP education. *IEEE Trans. on Education*, vol. 39, no. 2, pp. 120-126, 1996.

[7] M. Nahvi. Integration of DSP theory, experiments, and design: Report of a seven year experience with an undergraduate course. *Proc. ASEE Annual Meeting*, Charlotte, NC, 1999.

[8] C.H.G. Wright, T.B. Welch, M.G. Morrow, and W.J. Gomes, III. Teaching real-world DSP using Matlab and the TMS320C31 DSK. *ASEE Comput. Educ. J.,* vol. X, pp. 28-35, 2000.

[9] J. Orr and G. Stannard. The role of project work in the EE curriculum: Experience at Worcester Polytechnic Institute. *IEEE Trans. on Education*, vol. 22, no. 2, pp. 99-104, 1979.

[10] H. Hakim, ECE Department, Worcester Polytechnic Institute, personal communication.

[11] J.M. Wilson and W.C. Jennings. Studio courses: How information technology is changing the way we teach, on campus and off. *Proc. IEEE*, vol. 88, no. 1, pp. 72-80, 2000.

[12] R. Chassaing and D.W. Horning. *Digital Signal Processing with the TMS320C25*. New York: John Wiley & Sons, 1990.

[13] R. Chassaing. *Digital Signal Processing with C and the TMS320C30*. New York: John Wiley & Sons, 1992.

[14] R. Chassaing. *Digital Signal Processing: Laboratory Experiments Using C and the TMS320C31 DSK.* New York: John Wiley & Sons, 1999.

[15] D. Hatley and I. Pirbhai. *Strategies for Real-Time Systems Specification*. New York: Dorset House, 1987.

[16] L. Schachterle, Provost's Office, Worcester Polytechnic Institute, personal communication.

[17] P. Davis, M. Dimentberg, J. Goulet, J. Miller, S. Pierson. *What Makes for a Successful Bridge? An Interim Report of the NSF Institution Wide Reform Project.* June, 1999.

Acknowledgements

The author gratefully acknowledges the support of the Texas Instruments University Program (which donated all hardware and software for the studio laboratory) and the WPI Educational Development Council (which provided partial support for the development of the course). The author would also like to thank Profs. R. Chassaing, H. Hakim, and J. Miller for their interest and encouragement.

Biography

NATHANIEL A. WHITMAL, III,

Nathaniel A. Whitmal, III is currently an assistant professor at Worcester Polytechnic Institute in Worcester, Massachusetts. He received the B.S. in Electrical Engineering from MIT, and the M.S. and Ph.D. in Electrical and Computer Engineering from Northwestern University. His research interests are in the area of digital signal processing, with applications in speech coding and enhancement, time-frequency analysis, electroacoustics, and hearing aid design.

Table 1: Syllabus for EE 3703

Date	Topics
Week 1	EE 2312 review: difference equations convolution, Z-transforms, DTFT. Introduction to structured analysis: data flow diagrams, process specifications, requirements dictionaries
Week 2	C31 architecture and instruction set. Stack usage. Conventions for C-callable assembly procedures.
Week 3	Programming of interrupts, serial port, and timers. Sampling theorem. Effects of aliasing and quantization.
Week 4	Implementation of FIR filters via circular buffering. Design of linear phase filters. Use of windows.
Week 5	Implementation of IIR filters. IIR filter design methods: impulse invariance, bilinear transformation
Week 6	Theory of discrete Fourier series. Derivation of radix-2 FFT algorithms.
Week 7	Implementation of FFT algorithms. Applications of the FFT.

Table 2: EE 3703 student grades

Student	Quiz Avg.	Lab Avg.	Final Avg.
1	6.80	38.50	50.30
2	8.78	99.25	94.65
3	7.73	95.45	88.17
4	8.55	94.11	90.67
5	8.00	94.61	88.77
6	9.88	99.25	99.05
7	9.78	91.03	93.72
8	6.93	95.68	85.11
9	9.33	89.48	90.99
10	9.35	95.86	94.92
11	7.85	94.11	87.87
12	7.23	95.68	86.31
13	9.50	91.03	92.62
14	8.78	95.45	92.37
Average	8.46	90.68	88.25
Maximum	10	100	100

Table 3: Questions for pre/post test

Question 1 (40 points)	Consider a discrete-time LTI system with impulse response $h[n] = \begin{cases} a^n, & n \geq 0 \\ 0, & n < 0 \end{cases}$ with input *x[n]* and output *y[n]*. a. Compute the Z-transform of this system's impulse response. b. Write a constant-coefficient difference equation which expresses the present value of *y[n]* as a function of past values of *y[n]* and past and/or present values of *x[n]*. c. What values of *a* may be used in a real-world implementation of this system?
Question 2 (60 points)	Write (in either ANSI C or C++) a subroutine (of return type void) that computes the convolution equation $y[n] = \sum_{k=0}^{N-1} h[k]\, x[n-k]$ for n = 0, 1, …, N-1. The input parameters to the subroutine should allow access to an input array *x[n]*, a filter array *h[n]*, an output array *y[n]*, and an array containing the values {*x[-p]*, *x[-p+1]*, …, *x[-1]*}, all defined in the scope of the calling program. The parameters *p* and *N* may be presumed to be globally defined constants.

Table 4: EE3703 student performance on pre/post-tests

	Pre-test			Post-test		
Student	Question 1	Question 2	Total	Question 1	Question 2	Total
1	3	5	8	6	25	31
2	0	15	15	2	25	27
3	0	0	0	0	30	30
4	0	37	37	8	35	43
5	0	25	25	11	24	35
6	2	33	35	2	32	34
7	0	0	0	0	0	0
8	0	8	8	20	35	55
9	40	45	85	40	45	85
10	0	37	37	3	29	32
11	0	18	18	16	34	50
12	1	46	47	20	32	52
13	0	37	37	20	42	62
14	6	41	47	8	42	50
Average	3.71	24.79	28.50	11.14	30.71	41.85
Maximum	40	60	100	40	60	100

Table 5: Number of students failing to receive partial credit on the pre/post tests

	Question 1	Question 2	Total
Pre-test	9	2	11
Post-test	2	1	3

Table 6: Grades in recommended background courses (and/or follow-up courses)

Student	EE2312	EE2801	EE3803	CS1005	CS2005
1	n	n	C	B	C
2	n	A	B	A	B
3	n	n	C	B	B
4	n	n	B	n	B
5	C	n	n	B	B
6	A	n	A	A	A
7	B	n	n	n	n
8	C	C	C	C	n
9	B	n	n	A	B
10	n	A	C	cr	C
11	n	B	B	cr	cr
12	B	C	C	B	n
13	n	n	n	n	A
14	A	n	A	B	A

EE 3803 = Microprocessor System Design
CS 2005 = Data Structures and Programming Techniques
“n” = not taken, “cr” = received placement credit

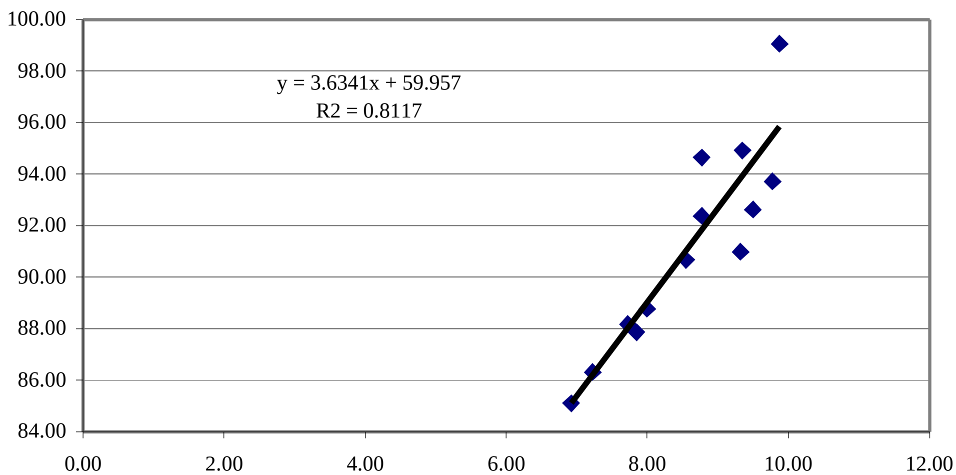

Figure 1: Quiz averages vs. final grades for Students 2 through 14