Digital Signal Processing in the Undergraduate Curriculum

Thomas L. Hemminger, Ralph M. Ford

Electrical and Computer Engineering
School of Engineering and Engineering Technology
The Pennsylvania State University at Erie, The Behrend College
Erie, PA

Abstract

The use of high-speed data acquisition and digital signal processing (DSP) technology has become the cornerstone of many areas of electrical engineering. This is particularly true in the fields of communications, controls, intelligent systems, signal and image processing. One of the driving forces behind DSP is the overwhelming interest in real-time processing as, for example, in high definition television, spread spectrum communications, and speech recognition systems. It is clear that DSP is instrumental in conveying the principles of many topics covered in electrical engineering particularly with respect to modeling and simulation.

The objective of this paper is to describe a multi-course sequence which employs DSP at many levels of the undergraduate curriculum for the purpose of enabling students to visualize, test, and implement concepts introduced in the classroom. This is accomplished through the completion of special projects and laboratory exercises in multiple courses with the goal of developing a solid foundation in engineering principles by the time of graduation. All students are required to take a core set of courses, which introduce DSP concepts, including applications where DSP is not typically employed. The level of complexity is increased as students progress through the curriculum, culminating in technical electives that extend their knowledge in a particular area of interest. The objectives of the sequence are realized through the employment of simulation tools and real-time hardware**.** This project is part of a plan to blend state-of-the-art technology with real world applications for the purpose of enhancing the undergraduate experience.

Introduction

For many years there has been a move to include discrete-time as well as continuous-time systems in electrical engineering curricula. This change has been driven largely by the availability of fast and inexpensive hardware. Therefore, colleges and universities have attempted to integrate digital signal processing (DSP) into many their courses. Our goal has been to introduce DSP in the 5th semester and to continue to build on this material each semester until graduation so that by the senior year students are well versed in filtering, modulation techniques, multirate sampling, and many other DSP topics. When possible we require students to work with discrete-time systems for controls, signals and systems, electronics, and

* A National Science Foundation ILI Grant #9850521 supported this work.

communications. We begin with the fundamentals of DSP in electronics I, followed by electronics II, and signals and systems. These courses provide a foundation in DSP concepts, requiring students to learn the software/hardware interfaces necessary to complete their projects and laboratory sessions. One of our main goals is to produce graduates that are knowledgeable in state-of-the-art technology. Another is to provide visualization methods for these students that enhance the learning experience and reduce the learning curve. With this in mind the faculty in electrical and computer engineering program (ECE) decided to thread DSP projects from the 5th semester to the 8th semester, requiring more rigorous experiments as students progress. To achieve this goal the ECE faculty has endeavored to introduce several platforms throughout the program so that students are well acquainted with software such as C++, MATLAB, and Hyperception. We feel that visualizing convolution, correlation, filter responses, FFT's and other DSP topics substantially enhances understanding of course material. An abbreviated course sequence is illustrated below:

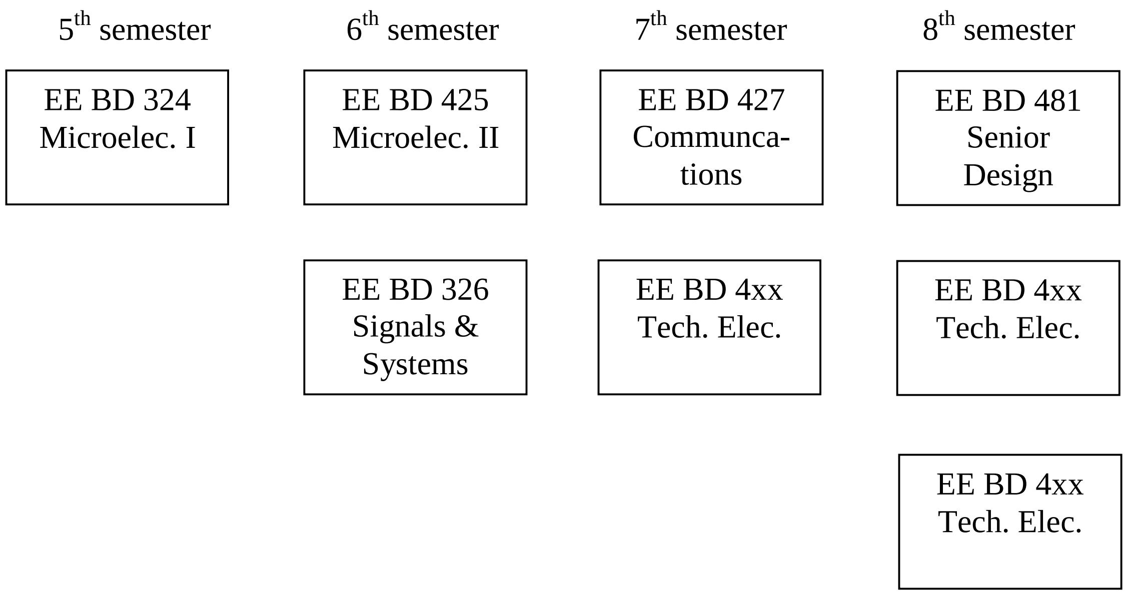

Figure 1: Semester timing of some of the courses related to this project. 4xx denotes electives.

The following software and hardware was utilized for this work:

- MATLAB with Signal Processing Toolbox, Signal Processing Blockset, Communications Toolbox, and Control Systems Toolbox from Mathworks, Inc [1]. This is a software tool for mathematical computation, analysis, visualization, and algorithm development. Until recently, it has not provided the capability to support real-time DSP processors.

- Hypersignal RIDE software development environment developed by Hyperception, Inc [2]. This is a graphical block diagram environment that allows rapid development of real-time DSP algorithms. It provides support for a variety of DSP hardware platforms. Eleven copies were purchased (8 for the laboratory and 3 instructor copies). This software allows rapid code generation for the target DSP processor without having to program it directly.

- Bridgenorth BN4002 DSP development boards [3]. These boards contain the Texas Instruments TMS320C40 32-bit 50/40ns Parallel DSP processor and have 32 Mbytes on-board memory capacity. They are coupled with BN3216, 16 bit A/D cards. A total of 8 board sets were purchased to equip each laboratory station. Hyperception RIDE supports real-time development for these boards.

- Texas Instruments (TI) TMS320C6211 DSK (evaluation) boards with the TMSC320C6211 processor were utilized. This is a more recent addition to the TI DSP processor line and provides higher performance than the TMS320C40 processor. Real-time application development on the DSK boards is more complicated than development using Hypersignal RIDE, as it requires a greater understanding of the hardware and C-like code development. Although code development using RIDE is simpler it is limited, and many DSP applications require the DSP engineer to develop at the code level and understand the hardware.

Course and Project Descriptions

1) EE BD 324 – <u>Microelectronics I and Laboratory</u>. This course is the first in a two course sequence in microelectronics and is taken in the first semester of the junior year by both electrical and computer engineering students. At this point the students have little exposure to DSP concepts and their utilization. The objective at this point in the curriculum is to introduce them to the concept of data acquisition and the use of DSP tools for data collection and processing. The Hyperception software and Bridgenorth data acquisition boards are utilized in one experiment in this class – the design of a power supply. The final testing requires that the ripple voltage be measured. Since the ripple voltage is quite low (2-10mV) it is difficult to accurately measure with traditional bench equipment. The data acquisition boards have 16 bit A/D converters over a 20V scale range. This provides a resolution 0.3mV, which is sufficient to resolve the ripple voltage. The intent is to have students utilize a Hyperception DSP workspace built by the instructor to collect and analyze the data.

2) EE BD 326 – <u>Signals & Systems and Laboratory</u>. This course introduces 6th semester students to digital signal processing (DSP) in a formal manner. The first half of the course deals with continuous-time (C-T) systems, using MATLAB to model linear systems in order to determine impulse response, step response, convolution, frequency response and most other (C-T) topics. The second half of the course consists mainly of discrete-time (D-T) topics related to their (C-T) counterparts. At this point students are confronted with simulations and real-time experiments. All students work in groups of two. There are approximately 10 laboratory experiments during the semester with 50% dealing with DSP. Some laboratories are intended to acquaint our students with the available software and hardware. Four significant experiments are emphasized here:

    a) Determination of sampling rate given a sinewave of unknown frequency (real-time). In this experiment students are required to observe a signal which is displayed on the computer screen when given the sample rate. This simple experiment emphasizes the fact that band-limited signals can be represented by a series of discrete samples. The samples are stored on disk, then the signal is reconstructed through the employment of low-pass filters, demonstrating how the superposition of their individual impulse responses affects reconstruction. Students quickly learn that signals must be band-limited before sampling to insure aliasing does not corrupt the signal. They also learn how to visualize the relationship between reconstruction in the time-domain and frequency-domain.

b) The infinite impulse response (IIR) filter (simulation). Students are required to develop low-pass, high-pass, band-pass, and notch filters and observe their effects on white noise. For this laboratory simulated data is employed and each student is required to determine the cutoff frequency based on the last four digits of their student number. The bilinear transform method is used which converts a real-time filter to a discrete-time filter at a specified center or cutoff frequency. This laboratory has been very successful in relating sampling, filter design, and difference equations in a unified fashion.

c) The IIR filter (real-time). Each student group is provided with a compact disk that contains music superimposed with a sinusoidal signal somewhere within the lower audio range. Students determine the frequency of the interference signal via FFT and develop a notch filter to eliminate the noise. This was a very successful laboratory experiment because it brought together concepts that students can directly relate to, i.e., music and filtering. The basic format is shown in figure 2 where in this case a 300 Hz sinusoid is superimposed on a music file. The lower display illustrates the spectrum of the noisy signal while the upper display shows the results after using a notch filter. The output is sent to a speaker/amplifier.

d) The finite impulse response (FIR) filter (real-time). Student groups are required to develop FIR filters based on the inverse discrete-time Fourier transform. This experiment is very similar to the IIR project described above except that the filter requires a much higher order than the IIR. Although the filter is numerically intensive it is emphasized that it has a linear phase response (odd number of taps) and is stable since it is non-recursive.

3) EE BD 425 – <u>Microelectronics II and Laboratory</u>. This is a 6^{th} semester course that is taken in the junior year and is required for electrical engineers, but is an elective course for computer engineering students. In this course, students develop complex electronic systems, one being the design and testing of a stereo graphic equalizer. The intent is to have students analyze the spectral content of signals for the design.

4) EE BD 427 – <u>Communications</u>. This is a 7^{th} semester course that does not incorporate a separate laboratory session, therefore students are required to complete projects on their own time and demonstrate the results to the instructor. Each student performs their own work, or work in groups of two persons depending on the complexity of the problem. At this point the course focuses primarily with simulation rather than real-time experiments because presently a laboratory session has not been assigned to this course. All of the

projects are conducted outside of class, thus making equipment security an issue. The ECE faculty is considering the addition of some laboratory time in order that students have access to real-time hardware interfaces and can be supervised by an instructor. Six experiments are currently offered, of which, students must choose 2 or 3, depending on the time available. All methods require modulation and demodulation. They are listed below:

- Double Side Band – Suppressed Carrier (DSB-SC).
- Double Side Band – Large Carrier (DSB-LC).
- Quadrature AM (QAM).
- Single Side Band – Suppressed Carrier (SSB-SC).
- Frequency Modulation (FM).
- Stereo (conversion from left & right channels to a single signal).

Two signals are always required, in order that they be combined onto one line to emphasize the advantages of frequency division multiplexing, or orthogonal properties as in QAM. Two experiments are illustrated in figures 3 and 4.

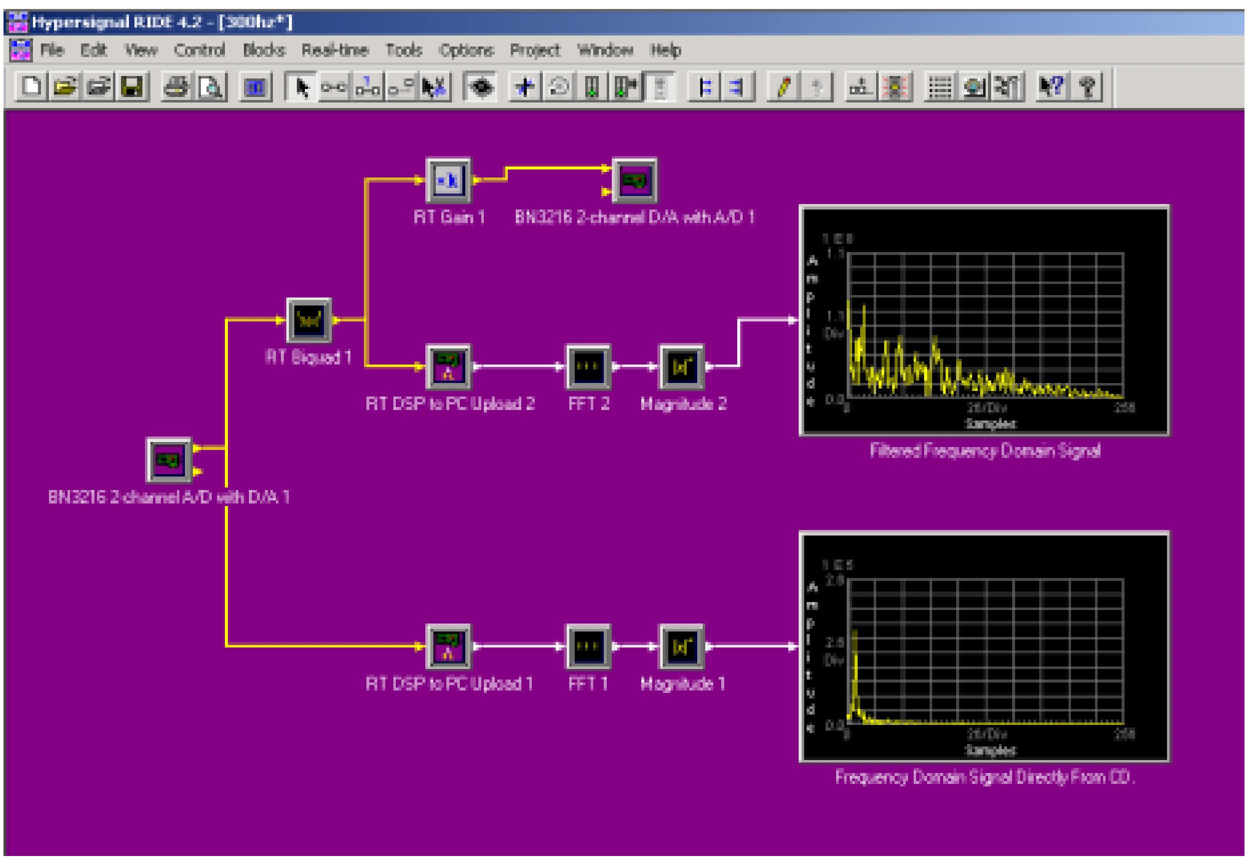

Figure 2: Real-time filtering of music superimposed with a sinusoidal tone. Lower display shows strong tone at 300 Hz while upper display illustrates music after filtering. Note, the displays are auto-scaling. The filters are second order and the sampling rate is 20 kHz.

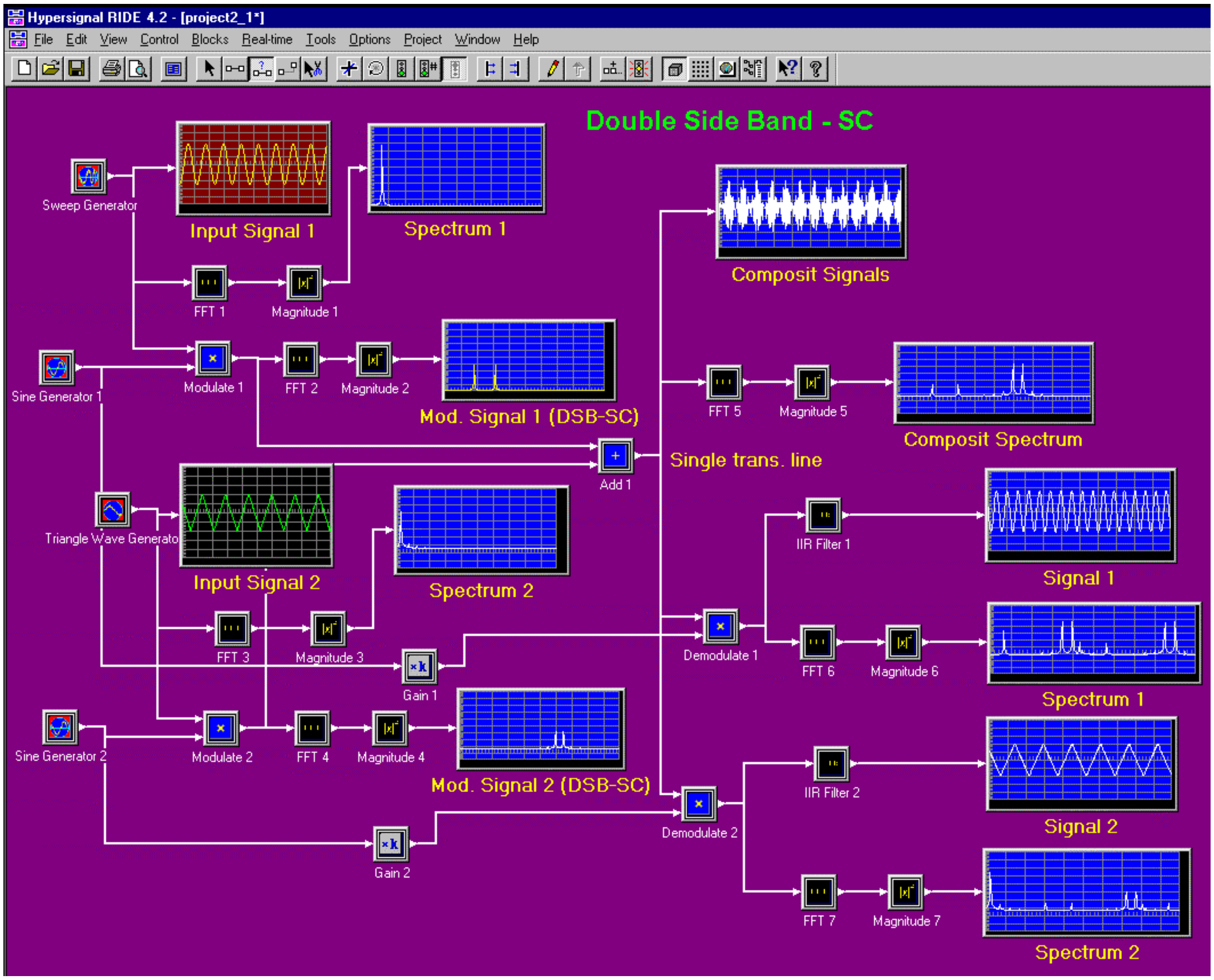

Figure 3: AM modulation illustrating time and frequency domain relationships.

All communications experiments require that the signal spectra be displayed for the purposes of determining the type of filters needed for demodulation. For example, the sweep generator oscillates at a maximum frequency of 300 Hz making the filter cutoff choice relatively simple. The triangle wave oscillates at 100 Hz but students must remember to include the harmonics making the cutoff approximately 1 kHz to retain sharp peaks. Figure 3 also illustrates the time delay experienced when processing a signal. This is apparent if the output of signal 1 is compared to the input, showing that the input frequency has decreased by the time the output has been displayed. Students choose the sampling rates and must explain all decisions.

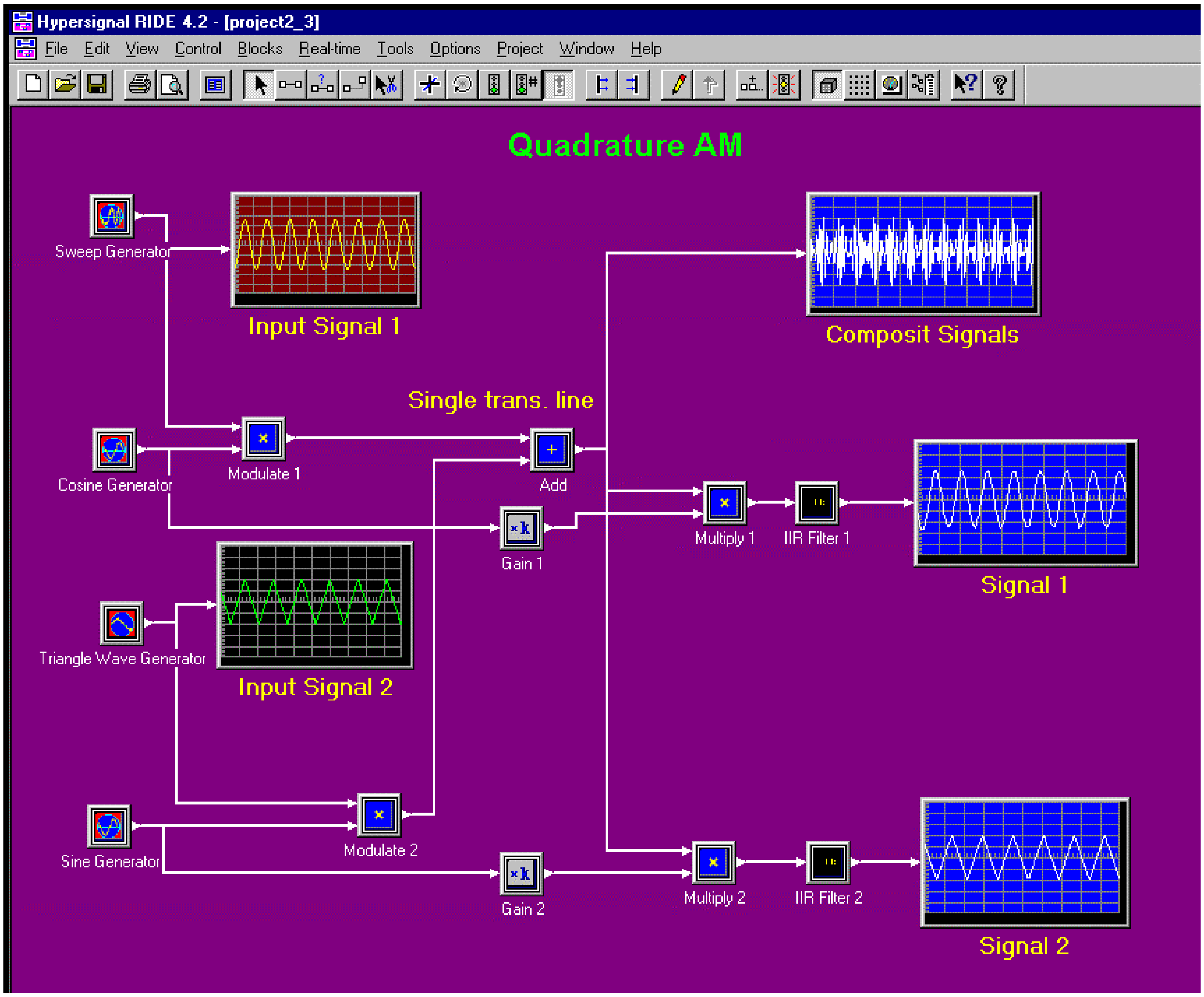

Figure 4: Quadrature modulation showing time domain relationships only.

6) EE BD 415 – Digital Control Systems. This elective runs almost every year in the 7th semester. It does not incorporate a separate laboratory session, therefore students are required to complete projects on their own time and demonstrate the results to the instructor. Students work in groups of two. In this course it is emphasized that digital control systems should be designed in the discrete domain using the zero order hold equivalent of the plant. The resulting controller can be written as a difference equation or a set of equations if state-space control is used.

   a) Motor controller. In this case a small DC motor was controlled digitally by sampling the rotation rate and using a computer to compensate for errors in RPM under different loads. In this case a motor employing an AC feedback winding was employed and rectified to DC. An RPM meter was used to calibrate the motor speed and the feedback voltage. A simple proportional controller was developed using a sample rate of 100 Hz. This same project was simulated using MATLAB's SIMULINK to verify the results. The students indicated that the software and hardware interfacing was beneficial to their learning experience.

b) Circuit response. In this lab students were provided a circuit that they were not permitted to analyze. They were then required to model it as a 2^{nd} or 3^{rd} order system by determining its step response. They developed an estimator and controller using state-space techniques in order to force a particular rise-time and overshoot. The required specifications were coupled with student numbers. This project turned out to be rather difficult due mainly to inaccuracies in the estimators.

7) EE BD 473 – Digital Signal Processing. This course is a senior level elective available to both electrical and computer engineers. The focus of the course is on both theoretical and applied aspects of DSP, and it is the capstone experience of the DSP education in the curriculum. In the prerequisite course, EE BD 326 (Signals and Systems), students are exposed to the basics of digital filter design (Z-transforms and basic FIR and IIR design). Furthermore, they already have experience with the filter design tools and real-time implementation capabilities available in the Hyperception environment and the Bridgenorth boards. Students are well-versed in the use of MATLAB at this point. In EE BD 473, MATLAB and Hyperception are utilized regularly for both in-class instruction and out-of-class projects. In addition, Texas Instruments TMS320C6211 DSK boards (evaluation boards with the TMSC320C6211 processor were utilized. Projects utilizing the starter kits required low level programming of the hardware for implementation. The objective was to have students develop competence in DSP at a variety of levels. This ranges from theoretical to applied designs. The DSP experience that students gained in previous courses allowed them to complete advanced projects in this class. Several different instructors have taught the course, and the projects utilized have varied from semester to semester. Projects that students have completed include the following:

a) *Sample Rate Conversion.* Sample rate conversion is an important topic in DSP, but difficult to grasp. It is important because in practice digital systems that operate at different sample frequencies must be interfaced, and sample rate conversion must be utilized to facilitate this. It is difficult to understand because the sample rates are converted between systems (requiring interpolation and decimation) and students struggle to identify the correct spectrum and sample frequency at different points in the process. To address this, students were required to design and test complete decimator and interpolator systems. Furthermore, they explored the effect of sample rate conversion on the spectra of signals. Hyperception is an ideal tool for this. Sample rate conversion can be implemented with a single up- or down-sample block and the spectra easily plotted. Since the up- or down-sample rate can be easily changed during simulation, the effect on signal spectra is quickly fed back by examining the output spectra. This direct experimentation and observation allows students to identify when signal content will be lost. Examples of up-sampling and down-sampling of a sine wave are shown below.

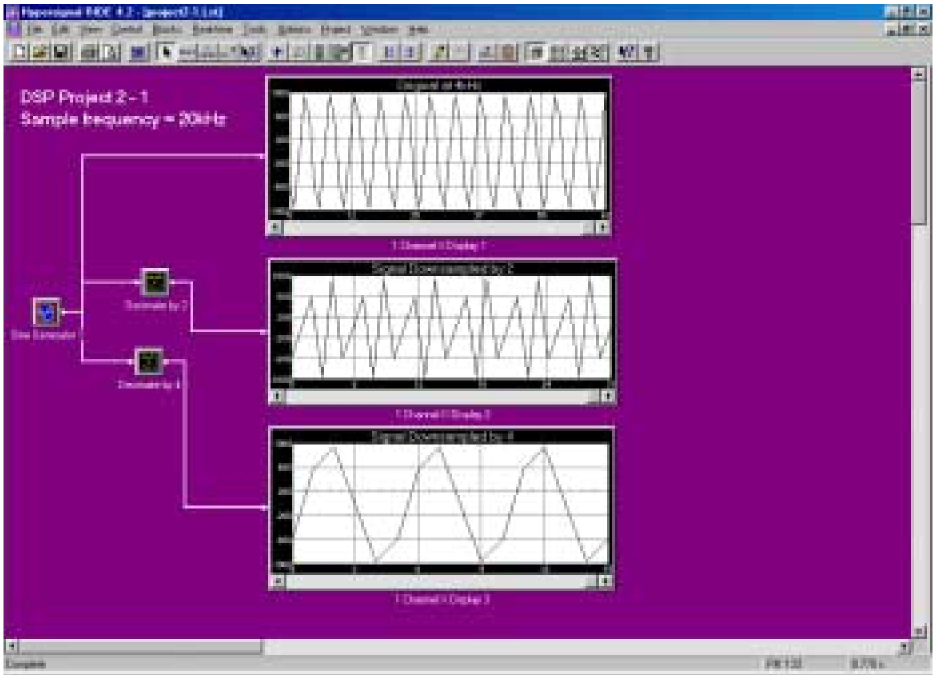

Figure 5. Downsamping of a sine wave.

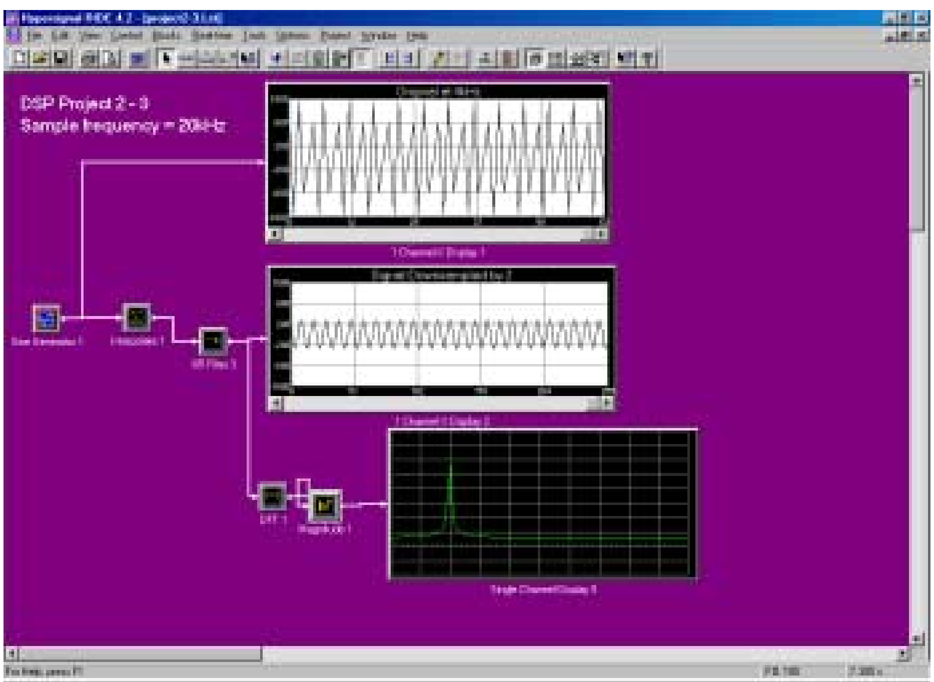

Figure 6. Upsampling of a sine wave.

After completing this first step of experimentation, students then design the interpolators and decimators. They were given the type of input signal and sample rates of the systems to be matched, but had to generate the design specifications on their own and develop and describe the rationale for their design. One possible solution to the design problem is shown below.

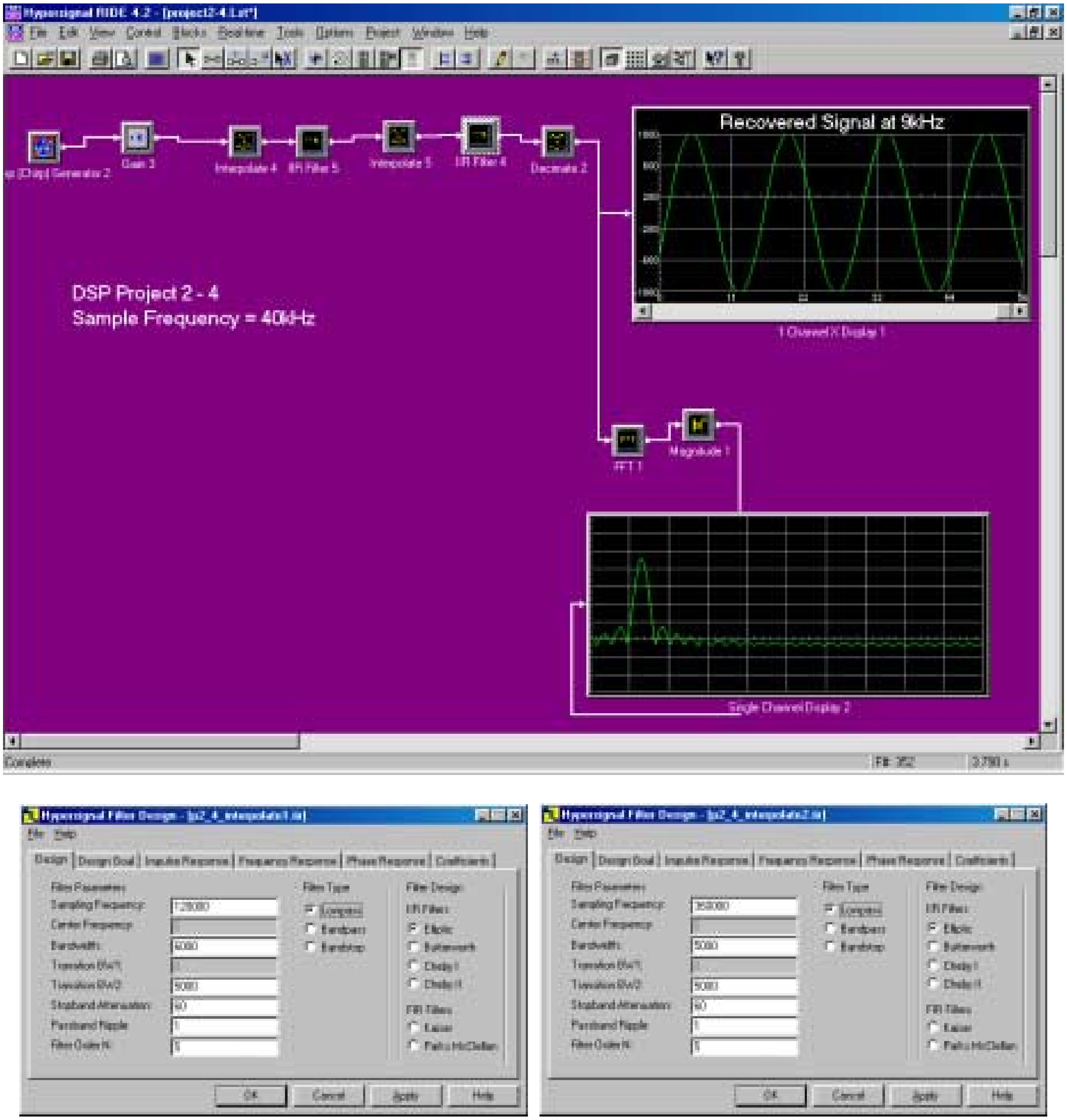

Figure 7. Sample rate converter design.

Overall, student feedback indicated that this was a valuable way to improve their comprehension of sample rate conversion and it also allowed them to develop a real solution to a given sample rate conversion problem.

b) *Design of a 8-band graphic digital graphic equalizer*. This project required students to design an 8-band graphic equalizer and implement it on DSP hardware via two methods. First, the equalizer was designed using the Hyperception graphical interface and simulated in real-time on the Bridgenorth boards. This required students to understand the high-level structure equalizer and be able to design and integrate the individual components. The Hyperception development environment allows this to be developed relatively rapidly, without requiring a tremendous command of DSP hardware implementation details.

After this was completed, the students were then required to implement the design on the TI DSK board. This requires them to actually write the processor code (the instructor supplied a code framework as a starting point) to implement the processing. The filter coefficients computed from Hyperception were utilized in the implementation. The resulting product has the same functionality, however, this task requires a low-level in-depth knowledge of DSP algorithm implementation. Thus, it requires students to develop an understanding of practical implementation at a lower level, rather than the abstract block diagram approach that Hyperception provides.

c) Other projects. There were a variety of other projects that were implemented as well:
- Spectral Estimation. Examination of spectral estimation, windowing, resolution of closely spaced sine waves, and the Short-Time Fourier Transform.
- Communication filter simulation. Narrowband noise generation, removal, and examination of filter tradeoffs.
- Correlation and a communication systems decoder.

In summary, integration of DSP in the curriculum in previous courses has provides a solid base for this capstone course and allows the development of advanced applications and real-time hardware implementations.

8) EE BD 480 & 481 – <u>Electrical Design Projects and Capstone Project</u>. This is a two course sequence in the 7th and 8th semesters. Many of our students work with software for the development stage but do not necessarily use DSP for their final project, although we certainly encourage it. Projects using a significant amount of DSP are included below:

a) *Noise Cancellation*. A local corporation in Erie, PA oversaw this project. These students were requested to acquire an acoustic signal and cancel it by destructive interference. They sampled the signal at approximately 50 kHz then developed a controller to create a signal 180° out of phase from the original. At certain locations the noise was reinforced by constructive interference or cancelled by destructive interference. Our students learned a great deal about acoustics and felt that this was a worthwhile learning experience.

b) *Digital Aircraft Auto Pilot*. For this project a wind tunnel ($\approx$40 ft./sec.) was incorporated with a specially designed airplane tail section placed inside. The objective was to create a digital control system to stabilize the pitch and yaw of the craft. Variable resistors were used as transducers. Students were required to meet a certain steady-state error and overshoot. After creating a system model, a digital controller was developed to meet the required specifications. Results were quite good considering the high wind speed and turbulence.

c) *American Sign Language*. Research in this area has been performed previously by many others, but our faculty determined that it would be a challenging undergraduate project. This group designed a glove employing strain gages that measured voltage when flexed for the 26 letters of the alphabet. These data were input to a DSP board and segregated using an unsupervised neural network. This resulted in five data clusters, which were used to individually train a supervised network employing MATLAB. After the weights were determined the group wrote a feedforward network in Visual Basic and developed a real-time sign language program that was highly successful. Only two or three letters presented a problem but the remaining characters were recognized immediately. The sample rate was 100 Hz so that data from each position of the hand could be averaged, thereby reducing the effects of noise. The reason for choosing multiple networks was two-fold; 1) Training the entire alphabet proved to be very difficult; 2) Each small network could be trained individually without affecting other networks.

d) *Vision Guided Autonomous Vehicle.* A vision guided autonomous vehicle was developed to navigate an environment using visual markers. To complete the project, the students (a team 5) developed an image acquisition system (lens, camera, data acquisition board) and an on-board computer for the signal and vision processing algorithms. They team had to apply their signal processing knowledge to understand edge detection algorithms. They successfully implemented a rudimentary vision algorithm and were able to have the vehicle navigate by following a tape marker on the floor.

e) *MP3 Decoder.* The objective of this project was to develop an MPEG layer 3 real-time decoder. The system was a proof-of-concept to show that streaming MP3 audio from the computer serial port could be sent to a decoder circuit, be successfully decompressed, and then be played on a speaker. In order to develop this, the students needed to first develop an understanding of the MP3 decoder and Discrete Cosine Transform (DCT) that is utilized in the compression scheme. The DCT is not a subject that is typically covered in the curriculum.

f) *Inverted Pendulum*. This is a well established problem although undergraduates can find it very challenging. These students developed a single axis rail system using model train tracks and a flat bed train car. They used a variable resistor to determine position and input this into a DSP board. They also wrote C-code for the controller. The main problem was with the torque calculations for the motors. The starting torque available from the motors was much lower than expected. The result was the addition of an additional motor to increase torque. This translated into a system that could maintain a vertical orientation when disturbed by about 20 degrees.

8) Research projects. Undergraduate students in the program are encouraged to participate in research projects with faculty. The projects listed below required a solid understanding of DSP and imaging fundamentals and the students were able to contribute to the research. The results of some of these projects are listed below:

   a) *Video shot boundary detection.* A current research problem is to automatically identify shot boundaries in digital video sequences. This is necessary for automatic catalog and retrieval of large amounts of video data. This work focused on developing new metrics for measuring the similarity of images in sequential digital images and comparing the results to existing metrics. The students were required to research and understand relevant literatures, work with the faculty member to develop new metrics, and implement (in C) and test the metrics on video data. This work led to the development of several new metrics, and the results have been published [4].

   b) *Video Shot Boundary Classification.* This research extended the previous and extended it to handle the detection of more complex shot boundaries, and furthermore, developed a method to classify them utilizing Fuzzy Logic Principles. One undergraduate student researcher was employed on this project to investigate different fuzzy implementations. which resulted in a publication [5].

   c) Synthetic Aperture Radar (SAR) Image Analysis. This research addressed the detection of changes in SAR imagery for military applications. The student researcher on the project had to utilize his DSP knowledge and extend it to understand image processing and analysis algorithms. Under the supervision of a faculty member he implemented and evaluated a variety of imaging algorithms for this task. The results of this work have been published [6].

Conclusions

DSP can be introduced carefully into the curriculum earlier than is typically done in the senior year. We have successfully introduced fairly advanced DSP concepts in the junior year including FIR and IIR filter design. Introducing DSP tools and concepts earlier provides a benefit in that the software and hardware tools can be used in multiple courses (Signals, Control Systems, Communication Systems, Digital Signal Processing). MATLAB, which is commonly used in many curricula, is an example of this. However, the use of a real-time development system, provides the advantage of allowing rapid real-time DSP systems to be created that has not been available in MATLAB until recently. This allows students to experience real-time DSP applications earlier in their academic career than is typical. This work has successfully shown that advanced DSP projects can be carried out by students at the undergraduate level as evidenced in capstone design projects and student research projects. Formal mathematical development of DSP principles is still appropriate for coverage at the senior level. Although, our students experience DSP (theory and applications) quite a bit in the junior year, they need the formal mathematical development in to address the theoretical and DSP design issues in the senior course (EE BD 473 – Digital Signal Processing). In the senior year applications that involve programming the DSP board itself can be developed. Over the past several semesters we have noted a marked improvement in our students' ability to grasp signal processing concepts.

References

1. MATLAB, The Mathworks Inc., Natick, MA

2. Hypersignal RIDE, Hyperception Inc., Dallas, TX.

3. Bridgenorth Signal Processing Inc., Blaine, WA.

4. R.M. Ford, C. Robson, D. Temple, and M. Gerlach, Metrics for Scene Change Detection in Digital Video Sequences, ACM Multimedia Systems Journal, January 2000.

5. R.M. Ford, A Fuzzy Logic System for Shot Boundary Detection in Digital Video Sequences, SPIE Conference on Storage and Retrieval for Image and Video Databases VI, February 1998.

6. F. Ahmed, B. G. Swahn, and G. J. Power, Multiresolutional quality assessment of SAR azimuth sequence using a spatio-temporal analysis technique, SPIE Conference on Capture, Analysis, and Display of Image Sequences II, January, 2000.

THOMAS L. HEMMINGER
Thomas Hemminger earned his Ph.D. degree in electrical engineering from Case Western Reserve University in Cleveland, OH. He is an Associate Professor in the electrical and computer engineering program at Penn State Erie and has been on the faculty since 1992. He is also a member of IEEE. His areas of expertise are neural networks, communications, digital signal processing, and computer vision.

RALPH M. FORD
Ralph M. Ford received his Ph.D. in electrical engineering from the University of Arizona in 1994. He has been a faculty member in Electrical and Computer Engineering at Penn State Erie since 1994, and holds the rank of associate professor and chair. His research interests are in the areas of signal processing, pattern recognition, and computer vision. He is a member of IEEE and the Optical Engineering Society.