Session 1626

Undergraduate Design Projects in a Laboratory for Real-Time Signal Processing and Control

Richard J. Kozick
Bucknell University

Abstract

A laboratory containing digital signal processing (DSP) units and computer workstations has recently been established at Bucknell University. The DSP units are programmed and controlled through a graphical interface on the workstations. The graphical interface provides an integrated environment for simulation and real-time implementation of signal processing and control algorithms. The laboratory is well-suited for undergraduate student design projects, and several such projects are described in the paper.

1 Introduction

An integrated laboratory for real-time signal processing and control has been established at Bucknell University. The laboratory has ten digital signal processing (DSP) units from dSPACE Corporation [1] and ten Sun workstations. The dSPACE units are controlled and programmed through Simulink [2], which is a graphical interface to Matlab [3]. The dSPACE/Simulink combination provides an integrated environment for modeling, simulation, and real-time implementation.

The typical mode of operation in the laboratory is to first develop a block diagram in Simulink, then run a simulation on the Sun workstation, and finally to download a real-time implementation for execution on the dSPACE unit. Only a minor modification is required to convert a block diagram from the simulation module to the real-time implementation. Simulink allows parameter updates on-the-fly as the program runs on the dSPACE unit. The dSPACE/Simulink environment facilitates rapid prototyping of real-time algorithms and allows "what if" investigations into the effects of parameter variations. The graphical interface shields students from low-level details so they can focus on higher-level design issues. The close relationship between simulation and real-time implementation makes the laboratory ideally suited for undergraduate student design projects. Parameters are varied easily in both the simulation and the real-time implementation, so simulation results can be matched with experimental results and design specifications can be achieved.

The laboratory environment has been used in several undergraduate electrical engineering courses, as well as in the first-year "Exploring Engineering" course that includes students from engineering and the liberal arts and sciences. This paper describes some particular student-chosen design projects in the areas of digital audio and speech processing, telephone tone detection, electrocardiograph processing, communication systems, and control systems.

The paper is organized as follows. The laboratory hardware, software, and operation are described in Section 2. Several undergraduate design projects are described in Section 3, and Section 4 contains concluding remarks.

2 Laboratory Description

The laboratory hardware and software are described in Section 2.1, and Section 2.2 contains a simple example that illustrates simulation and real-time implementation.

2.1 Hardware and Software

The main hardware components are ten dSPACE DS1102 Miniboxes and ten Sun workstations. The dSPACE Miniboxes and Sun workstations communicate with each other through an Internet connection. Each Minibox and workstation has its own IP (Internet Protocol) address and is remotely accessible from any Internet site. The Internet access to the laboratory makes it possible to develop experiments that can be shared among several universities, as proposed in [4] and recently developed further in [5].

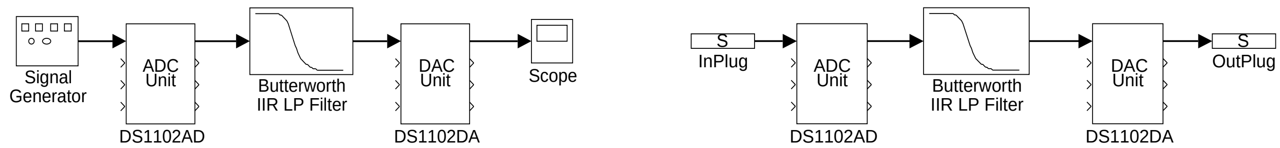
Each DS1102 Minibox contains a 40 M Hz Texas Instruments TMS320C31 digital signal processor along with memory and input/output circuits, including four channels each for analog-to-digital and digital-to-analog conversion. The dSPACE Miniboxes can be programmed in assembly language, C language, or through Simulink, which is a graphical interface to MATLAB. The Simulink interface is the simplest to use, since systems and algorithms are described by block diagrams. Simulink contains an extensive library of predefined blocks for signal and system analysis.

An additional software tool called TRACE serves as a virtual oscilloscope for real-time data collection and display. TRACE can be used to transfer measured data directly to MATLAB for analysis. TRACE is also a valuable aid for debugging algorithms and tuning parameters during real-time execution on the Minibox. The combination of Simulink, MATLAB, and TRACE provides an integrated set of tools for simulation, real-time implementation, and testing of signal processing and control algorithms.

2.2 Simulation and Real-Time Execution

Once an algorithm or system is described by a block diagram in Simulink, it can either be simulated on the Sun workstation, or compiled, downloaded, and executed in real-time on a dSPACE Minibox. The simulation runs on the workstation only, while the real-time algorithm is executed on the dSPACE Minibox and is used to process external signals and connect to physical systems. The real-time version can be developed rapidly from the simulation version, as described next.

A simple example to illustrate the easy migration from simulation to real-time implementation is shown in Figure 1. Figure 1a is the block diagram for simulation of a low-pass digital filter, and Figure 1b is the block diagram for real-time execution of the same digital filter. Both diagrams contain analog-to-digital converters (ADC) and digital-to-analog converters (DAC) that sample the signals in time and quantize the sample values. Note that four channels are available on the ADC and DAC. The only difference between the two diagrams is that the input signal in Figure 1a is a computer-generated signal for the simulation, while the input signal in Figure 1b is a *real* signal that is physically connected to the dSPACE hardware. This connection is represented by a *plug* in Figure 1b. Similarly, the output in



(a) (b)

Figure 1: (a) Simulink block diagram for simulation of a low-pass digital filter. (b) Simulink block diagram for real-time implementation of same digital filter on dSPACE hardware.

Figure 1a is a simulated oscilloscope, while the output in Figure 1b is a plug that signifies the connection to a physical hardware device, such as an oscilloscope or audio speaker. All of the processing in the real-time implementation must occur between the ADC block and the DAC block, since this represents the data available inside the dSPACE Minibox.

Any simulation block diagram is converted to a real-time block diagram using the same principle of adding plugs to denote the input/output of real signals to/from the dSPACE hardware. The close connection between simulation and implementation encourages students to iterate back and forth between the two modes. For example, a system model in the simulation may be revised in response to real-time testing with an actual system. Iteration of this type improves the system model, and also provides the student with greater understanding of the system and the modeling process. Similarly, once an algorithm is developed and tested in the convenient framework of the simulation, it usually works as expected in the real-time implementation. This facilitates rapid prototyping of real-time algorithms.

An interesting feature of the interface between the Simulink software and the dSPACE hardware is on-the-fly updating of parameters during real-time execution. For example, suppose the cutoff frequency of the digital filter in Figure 1b is changed in the Simulink diagram while the filter is executing on the dSPACE hardware. This change takes effect in real-time on the dSPACE hardware, without recompiling or restarting the filter. This capability is useful for parameter tuning during real-time execution.

3 Design Projects

The laboratory has been used in several ways since the fall of 1994. More than 200 students from all engineering disciplines and the liberal arts have been introduced to basic concepts of signal analysis and digital signal processing in the “Exploring Engineering” course [6]. The laboratory is also used for classroom demonstrations in several courses to illustrate sampling, aliasing, digital filtering, and modulation.

Electrical engineering students in the junior year at Bucknell University have used the laboratory to develop design projects in the “Signals and Systems” course during 1994 and 1995. The project spans the final four to five weeks of the semester. Students are expected to choose a topic for the project, design an appropriate system model or algorithm, simulate the design, and implement the system in real-time to process actual signals or control a real system.

During 1994, the real-time implementations were programmed in C language, since the graphical programming interface was not yet available. Most projects in that year were limited to one or more digital filters with some simple logic. The digital filters were designed with Matlab and manually implemented in C. Example projects included a digital crossover network for audio speakers, digital bandpass filters to isolate harmonics in a signal, a detector for a subset of telephone touch-tones (DTMF), digital demodulation of AM signals, and a digital guitar tuner.

During 1995, the availability of the graphical programming interface produced a considerable improvement in the level of the projects. The freedom from C programming allowed students to focus more attention on system design and signal processing issues. The following subsections describe selected design projects from the 1995 course.

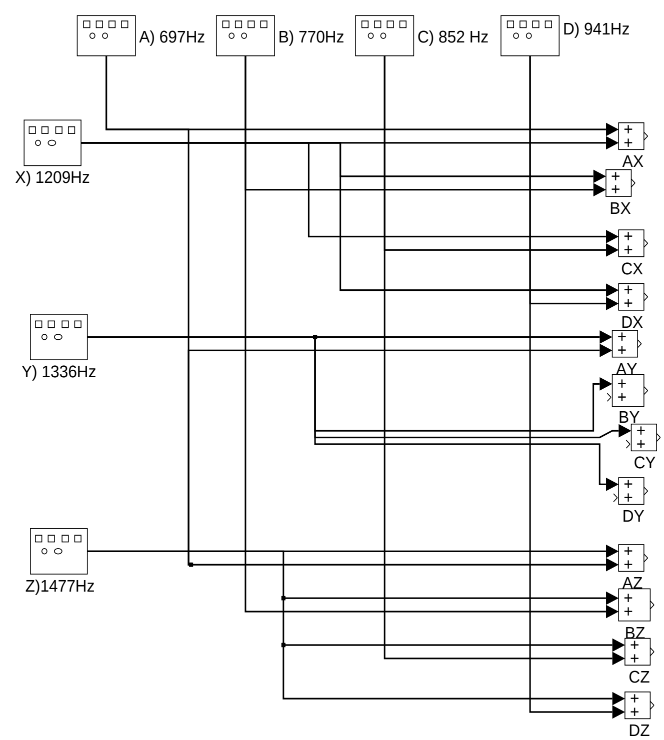
3.1 DTMF Detector

The sounds generated by a touch-tone telephone keypad are each composed of two frequencies. The dual-tone multi-frequency (DTMF) code is summarized in the table of Figure 2a.

One student successfully implemented a fully-functioning DTMF detector. The project nicely illustrated the proper use of all the features of the development environment. The DTMF detection system was first designed and simulated with Simulink. The Simulink block diagram that generates the DTMF tones for simulation is shown in Figure 2b. It consists of sine wave generators and summers to add pairs of sine waves to produce the twelve combinations in Figure 2a.

Frequency (Hz)	1209	1336	1477
697	1	2	3
770	4	5	6
852	7	8	9
941	*	0	#

(a)



(b)

Figure 2: (a) DTMF frequencies for telephone keypad. (b) Simulink block diagram for generating DTMF signals for simulation.

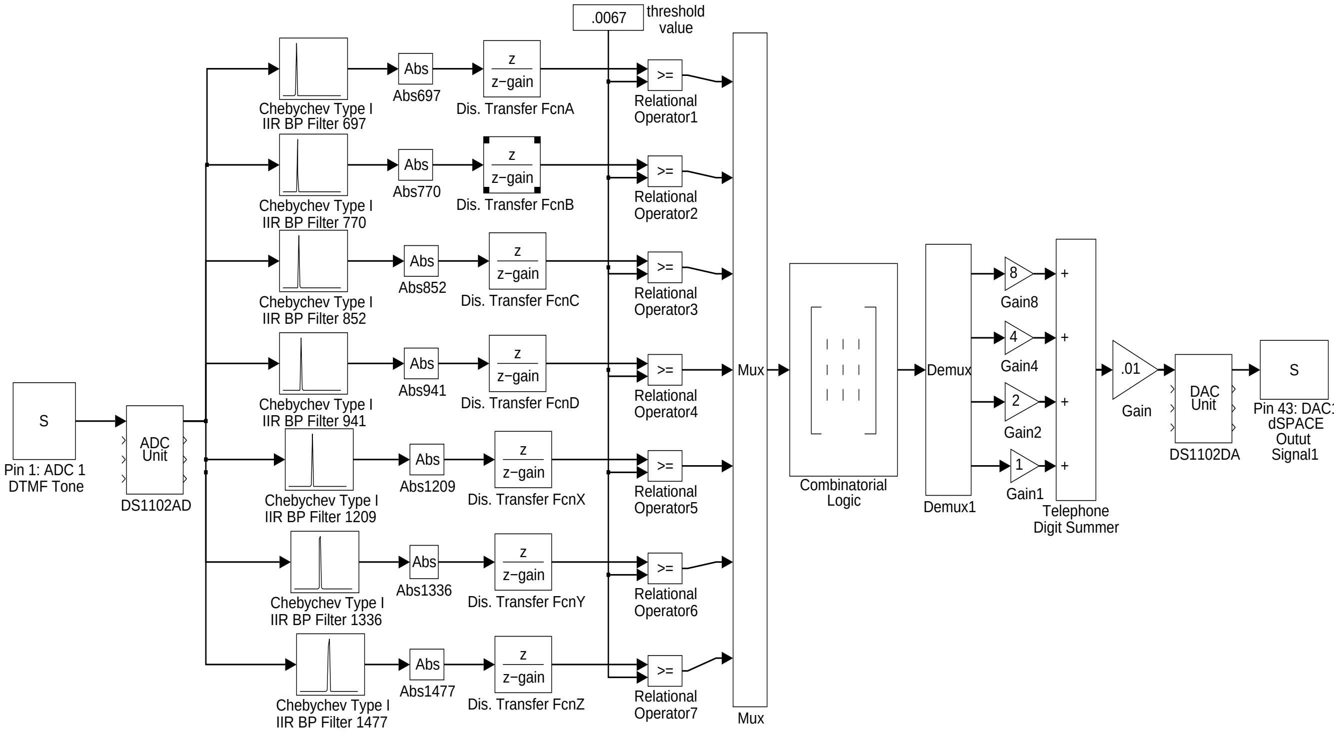


Figure 3: Simulink block diagram for real-time implementation of DTMF detector.

Figure 3 contains the DTMF detector for real-time operation. The same detector is used for simulation by connecting a signal from Figure 2b to the ADC Unit in place of the plug labeled "S". The detector first filters the input signal by a bank of seven band-pass filters tuned to each of the DTMF frequencies in Figure 2a. Then the output energy from each filter is estimated and compared with a threshold. The results of the threshold comparisons are applied to a decision algorithm that identifies the corresponding telephone key.

The simulation environment was convenient for designing, testing, and debugging the algorithm. The effects of filter order and bandwidth were easy to evaluate via simulation. Once the simulation performed properly, it was converted to the real-time implementation shown in Figure 3. The TRACE tool was used in conjunction with on-the-fly updating of parameters to determine the proper threshold for real-time implementation. The real-time system performed well with actual DTMF signals from a telephone.

I found it remarkable that this fully-functioning system was completed by a single student in a short amount of time. The focus of the project was on system-level issues rather than low-level programming details, which is appropriate for the signals and systems course.

3.2 Digital Audio Equalizer

A group of three students developed a five-band, adjustable digital audio equalizer. The Simulink block diagram for real-time implementation is shown in Figure 4. The blocks labeled 750 Hz, 1.5 kHz, 2.5 kHz, 3.5 kHz, and 4.2 kHz are adjustable gains for the frequency bands. As these gains are changed in

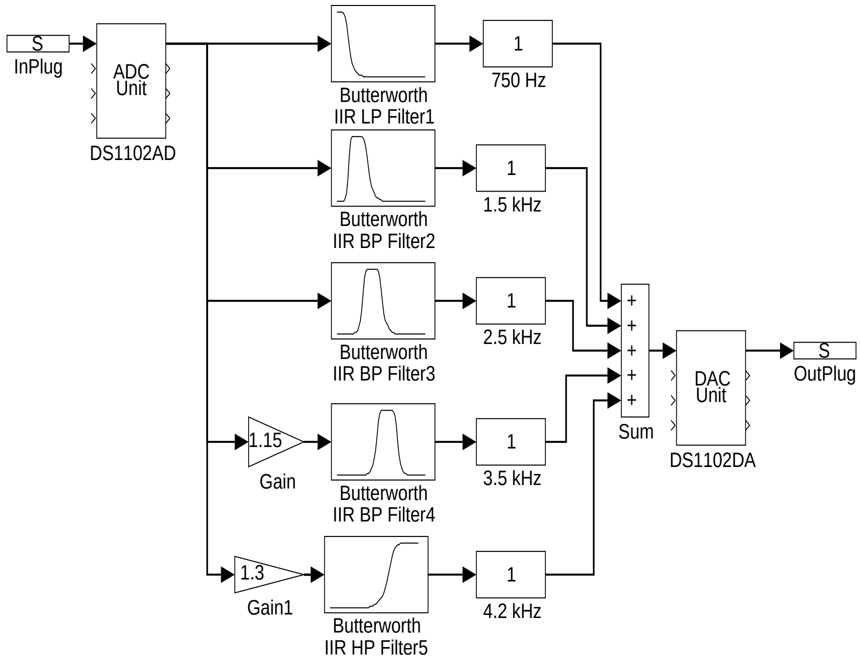


Figure 4: Simulink block diagram for real-time implementation of digital audio equalizer.

the block diagram, the real-time algorithm is updated on-the-fly and the effect is observed in the output signal. The system was demonstrated with real music waveforms.

This project contained several interesting design issues. First, the digital filter type, order, and frequency range had to be chosen. Then, analog filters were designed for anti-aliasing (at the dSPACE input) and reconstruction (at the dSPACE output). Students were somewhat surprised that the first- and second-order analog filters in their textbook were not sufficient for this application, so they designed higher-order filters. The students also characterized the frequency response of their overall system using a computer-controlled GPIB frequency response measurement system that they developed earlier in the semester. Interesting effects were observed in the overall response at frequencies where the equalizer bands overlap. The phase response of each filter is important in these overlap regions, and the challenge of designing of an overall "flat" response for the equalizer was appreciated.

3.3 Audio Noise Removal

Audio signals received via short wave radio are characterized by impulsive noise as well as sustained background noise. A group of three students developed a system containing both linear digital filters and nonlinear median filters to remove noise from short-wave radio signals. The superior performance of the median filter in removing the impulsive noise was clear in the demonstration with real signals. The combination of median filtering with linear filtering was found to provide the best solution. The on-the-fly parameter update feature was useful in this system to switch between the linear filter and median filter in real-time, and also to demonstrate the effects of window size in the median filter and bandwidth in the linear filter.

3.4 System Modeling and Control

Several groups of students investigated system modeling, simulation, and real-time digital control using systems available at Bucknell University [7]. A detailed description of one such project is given in [8]. A pair of students developed a model and simulation for a servomechanism. The physical servomechanism was found to be nonlinear, and students appreciated the difficulties in constructing a model and realized some limitations of linear system theory.

3.5 Other Projects

Other design projects are briefly described below.

- Audio harmonizer: One student investigated algorithms for shifting the frequency of an audio signal up or down by an octave while preserving the duration of the signal. Algorithms were implemented and tested in Matlab and Simulink.

- Voice processing: Groups of students developed systems for distinguishing between the words ON and OFF, and for distinguishing between male and female voices. The algorithms were developed by students using linear techniques based on spectral and correlation analysis. These simple methods were not robust across a group of speakers, so the project illustrated some of the difficulties of automatic speech recognition.

- Modulation and demodulation: A group of three students developed a system that amplitude modulates a desired signal, transmits it over a channel along with other carrier frequencies, and then demodulates the desired signal.

- Signal averaged electrocardiography (EKG): A group of students interested in biomedical applications investigated digital filtering in signal averaged EKG. Actual EKG signals were downloaded from the Internet and processed in Simulink to illustrate the ideas.

- Acoustic echo processing: One student with an interest in sailing investigated methods for imaging the ocean floor. Simulations were developed in Simulink.

- Phase-locked loop (PLL) for carrier synchronization: This project was performed in the communication systems elective course. The objective was to design and simulate the PLL in Simulink, then implement it in real-time on the dSPACE hardware.

4 Concluding Remarks

Undergraduate design projects using a laboratory for real-time digital signal processing and control have been described. Project topics were chosen by students and performed as part of electrical engineering courses at Bucknell University. The graphical programming interface and integrated environment for simulation and implementation allow rapid development and facilitate a focus on high-level design issues rather than low-level implementation details. Sophisticated projects can be designed, simulated, and implemented in real-time within one month.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. DUE-9451618. The dedication of all students that worked on the projects described here is appreciated.

References

[1] dSPACE digital signal processing and control engineering GmbH, *DS1102 Software Environment* (Reference Guide), Germany, 1994.

[2] The Mathworks, Inc., *SIMULINK: Dynamic System Simulation Software* (Users's Guide), Natick, Massachusetts, 1992.

[3] The Mathworks, Inc., *The Student Edition of MATLAB*, Natick, Massachusetts, 1992.

[4] M.F. Aburdene, E.J. Mastascusa, and R. Massengale, "A proposal for a remotely shared automated laboratory," *1991 IEEE/ASEE Frontiers in Education Conference*, pp. 589-592, Sept. 1991.

[5] R.J. Kozick and M.F. Aburdene, "A dynamic parameter estimation experiment that is remotely accessible via Internet," *1996 ASEE Annual Conference*, Washington, D.C., June 1996.

[6] R.J. Kozick, "Electrical engineering laboratory for first-year and non-engineering students," *1994 IEEE/ASEE Frontiers in Education Conference*, San Jose, CA, Nov. 1994.

[7] J.W. Baish and E.J. Mastascusa, "Inexpensive experiments for undergraduate control system laboratories," *Proc. of the American Control Conference*, pp. 1596-1599, June 1988.

[8] R.J. Kozick and C.C. Crane, "An integrated environment for modeling, simulation, digital signal processing, and control," to appear in *IEEE Trans. on Education*, May 1996.

RICHARD J. KOZICK has been an Assistant Professor in the Electrical Engineering Department at Bucknell University since 1993. He was a Member of Technical Staff at AT&T Bell Laboratories from 1986 to 1989 and again from 1992 to 1993. He received the Ph.D. degree from the University of Pennsylvania, M.S. from Stanford University, and B.S. from Bucknell University. His research interests are in the areas of signal processing and communications.