

# AN INEXPENSIVE APPROACH FOR TEACHING ADAPTIVE FILTERS USING REAL-TIME DSP ON A NEW HARDWARE PLATFORM

Michael G. Morrow

Department of Electrical and Computer Engineering  
University of Wisconsin-Madison, WI

Cameron H. G. Wright

Department of Electrical and Computer Engineering  
University of Wyoming, WY

Thad B. Welch

Department of Electrical and Computer Engineering  
Boise State University, ID

## Abstract

Carefully chosen demonstrations and hands-on exercises of digital signal processing (DSP) concepts can result in improved comprehension and better retention of topics for the majority of our students. This is especially true when the instructor has limited time to present a specific topic. In this paper, we describe the in-class use of a demonstration using real-time DSP to implement a basic adaptive filter for noise cancellation, utilizing newly-available DSP hardware from Texas Instruments. We provide a description of the new hardware platform, how the demonstration was used, and the results from a short survey administered to the students regarding the demonstration.

## Introduction

An ability to apply digital signal processing (DSP) concepts to a variety of situations has become one of the key skills that many employers expect of new electrical and computer engineering (ECE) graduates. Many engineering educators have found that a true understanding of fundamental DSP topics can be more fully realized by students when they attempt to implement various DSP algorithms in real-time (typically in C), compared to implementing the same algorithms only in non-real-time using tools such as MATLAB or

LabVIEW [1, 2]. While both MATLAB and LabVIEW are excellent tools that have a necessary and valuable place in DSP education, extending student experience to real-time operation, and the associated pitfalls that go along with that, provides a real pedagogical benefit. In order to help students successfully transition from theory to real-time practice, there needs to be both a pedagogical method and an infrastructure in place to support them and target as many modes of learning as are reasonably possible.

The underlying concept of using demonstrations to enhance learning for DSP isn't new. Many engineering educators have recognized the need for, and written about the results of using interactive learning and demonstrations for this important subject area [3–7]. For many years, the authors of this paper have been suggesting and providing proven DSP teaching methodologies, hardware and software solutions, and DSP tools that have helped motivate students and faculty to implement real-time DSP-based systems to improve education in signal processing and related topics [8–20].

These efforts to promote the use of demonstrations and hands-on experiences for students have emphasized the fact that DSP is much more than just a collection of theories and problem solving techniques, and that hands-on experience with real-time hardware is extremely beneficial. Real-time

DSP not only helps students master various DSP concepts but also helps students retain their understanding of those concepts. That is, using real-time DSP as the catalyst, students can be more consistently motivated to explore and implement a wide variety of DSP topics in an environment in which they are limited only by their imagination.

After reading recommendations in the engineering education literature, talking with other professors, and trying various approaches on our own over the years, the preferred pedagogical approach favored by the authors is a three-step method of teaching DSP [20].

1. First, we teach the theory along with interesting and motivating real-time demonstrations, often using the highly versatile winDSK8 tool we've developed.
2. We then have students implement a particular concept using MATLAB, until they are comfortable with the basic topic.
3. Finally, we have them “de-vectorize” their MATLAB code with real-time operation in mind, and convert it to C, then compile and run it in real-time on high-performance DSP hardware.

This three-step method of teaching DSP has now been used by many educators at various institutions around the world. It has been found to get students past their initial trepidation with new theory, solidify their understanding of the new concepts within their own framework as aided by the demonstrations, allow practice at a fairly high level of abstraction (via MATLAB), then master the concept at a much more detailed level (by converting to C and making it all work in real time).

The freely-available winDSK tool has evolved over the years to support an ever-expanding number of DSP topics along with the latest in real-time DSP hardware. The current iteration, as of this writing, is winDSK8 which supports the most commonly available high-performance hardware used

for DSP education (discussed below). The main window of the graphical user interface (GUI) for winDSK8 is shown in Fig. 1.

There are several choices of hardware the professor may make if using this approach. For the very popular Texas Instruments (TI) processors, the Spectrum Digital C6713 DSK, the Logic PD Zoom OMAP-L138 Experimenters Kit (ZEK), and the relatively new Texas Instruments OMAP-L138 Low Cost Development Kit (LCDK) are all highly capable platforms currently available for real-time DSP, and all can be used effectively with the recommended pedagogical method. The LCDK is the newest of the three, having been introduced in the latter part of 2012. Its capabilities are equal to or greater than the other two platforms, and it is the lowest in cost (only \$195 as of this writing). One capability germane to this paper is the access to not only “line in” audio but also to “microphone in” audio that can support generic stereo microphones or dual monaural microphones. Note that in the winDSK8 GUI shown in Fig. 1, we have selected the LCDK as the real-time DSP board, chosen 48 kHz as the sample frequency, and selected “microphone in” as the audio input (highlighted in blue).

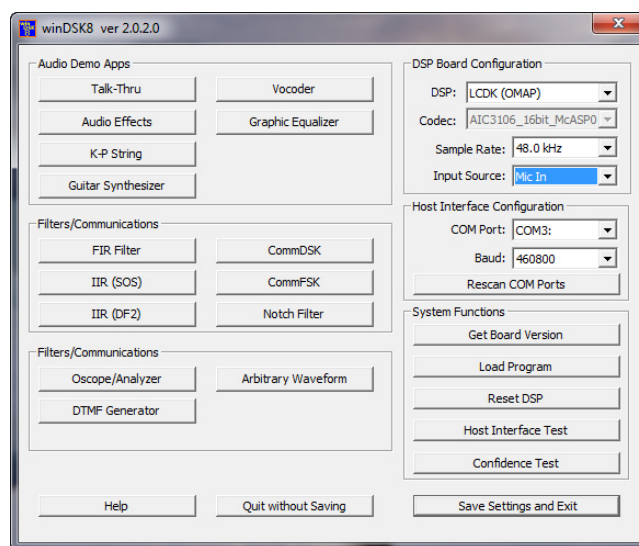


Figure 1: Main screen of winDSK8, with the “Mic In” input selected on the LCDK board.

## The Low Cost Development Kit for DSP

We have used many hardware platforms over the years to support our approach of incorporating demonstrations and hands-on experiences for DSP students. Since TI has been consistently supportive of DSP educators, we have almost always used their boards, using both fixed- and floating-point Texas Instruments (TI) processors, such as the C50, C31, C6201, C6211, C6711, C6713, and most recently the multi-core OMAP-L138 (which includes both a C6748 core and an ARM926 core). Of these processors, several are now only of historical interest, while the boards based on the C6713 and the OMAP-L138 remain our primary targets of interest.

As mentioned in the previous section, the newest board based on the OMAP-L138 is called the Low Cost Development Kit (LCDK) [21], shown in Fig. 2. A specific comparison of the LCDK with the C6713 DSK and the Logic PD Zoom OMAP-L138 Experimenters Kit (ZEK) can be found at [http://www.rt-dsp.com/2nd\\_ed/board\\_comparison.pdf](http://www.rt-dsp.com/2nd_ed/board_comparison.pdf). In general, the LCDK is superior in most ways to the C6713 DSK and equally (if not more) capable than the ZEK. For teaching purposes, the LCDK is our preferred choice. One of many reasons for that choice is that while the ZEK supports only audio “line in”

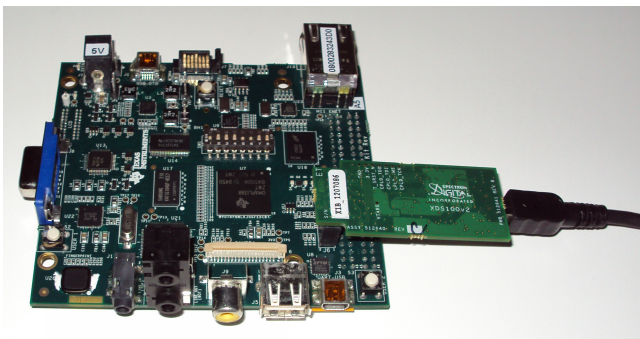


Figure 2: The new Low Cost Development Kit (LCDK) from Texas Instruments, with a multi-core OMAP processor and multiple I/O options. The XDS100 emulator board is shown on the right, plugged into the LCDK and connected to a USB cable.

and “line out,” the less expensive LCDK also supports “microphone in,” which allows more flexibility for demonstrations. Another capability the LCDK has, not available on the C6713 or the ZEK, is a user-configurable “video in” and “video out” that excites many students; this capability will be discussed in a future paper.

A more subtle advantage of the LCDK over the ZEK relates to the board design itself. While both OMAP-L138 boards use the identical audio codec chip, the two manufacturers chose to integrate that codec into the overall board design in different ways. In particular, the power supply decoupling of the codec chip is much better on the LCDK, and this is important to some applications. See the comparison on the [www.rt-dsp.com](http://www.rt-dsp.com) site mentioned previously for plots and a discussion of the noise related to this design difference. To be fair to both boards, it should also be noted that in order to program the LCDK in C (using Code Composer Studio from TI), you need an additional item: a small, inexpensive XDS100 emulator board that plugs into the LCDK, since this host interface isn’t built into the main board as it is with the ZEK.\* These emulator boards are available at a suggested retail price of \$79, from TI’s eStore or a variety of third-party vendors. The adaptive filter demonstration described in this paper was accomplished using the LCDK.

### Adaptive Noise Cancellation: Background

One application of this now-available stereo microphone capability, that is useful for teaching DSP to students, is adaptive noise cancellation using an adaptive digital filter [22–25]. In the basic block diagram shown in Fig. 3, two inputs (on the left of the figure) are provided to the system. The upper input contains “signal plus noise” while the lower input contains “correlated noise.” That is, the lower input is correlated with the additive noise present in the upper input, but *not* correlated with the desired signal. The adaptive filter (the box labeled with the

\*This small XDS100 emulator board can also be used with certain other DSP or microcontroller boards, if desired.

transfer function  $H_k(z)$  in the figure) uses the error signal to adjust its transfer function in real-time so as to optimally cancel out the noise at the output, leaving the desired signal with greatly reduced noise. The theory is straightforward, and will only be discussed briefly here.

Adaptive noise cancellation, as shown in Fig. 3, is one of many applications of adaptive filters. We make the common trade-off of accepting a higher filter order for guaranteed stability, and define the filter represented by  $H_k(z)$  to be an FIR filter. Hence, the filter coefficients also completely define the impulse response  $h_k[n]$ , which is simply the inverse  $z$ -transform of  $H_k(z)$ . These coefficients change over time in order to adapt the filter to changes in the noise, and thus the subscript represents the impulse response and transfer function at some sample time  $k$ . Note that since the filter coefficients change to adapt to the noise, we cannot guarantee linear phase response of the FIR filter unless we impose additional constraints of symmetry or anti-symmetry to the coefficients. This provides a nice opportunity to reinforce the concept of phase response with our students.

While it's implied in Fig. 3, we don't explicitly show a box for the particular update algorithm that changes the filter coefficients. There are many approaches to such an update algorithm, each with their own advantages and disadvantages. Two of the most well-known classes of algorithms for adaptive filters are the least mean square (LMS) and the recursive least squares (RLS) methods. While RLS converges more quickly than LMS and can thus better track rapidly changing noise, its much greater computational complexity often makes LMS the more attractive choice [22–25].

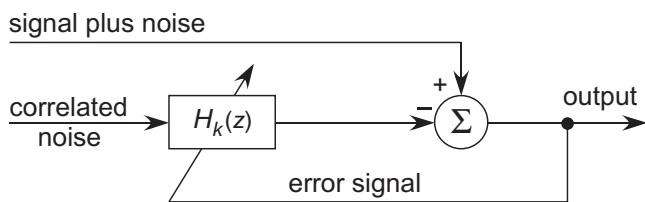


Figure 3: A basic block diagram of an adaptive filter used for noise cancellation.

## Adaptive Noise Cancellation: a Demo

For the scenario we present to our students, we are trying to duplicate the noisy environment of a firefighter at the scene of an emergency. The “signal plus noise” represents the combined signals from the firefighter’s helmet-mounted monaural microphone (where the “signal” is his/her voice, and the “noise” is from a chainsaw running in the background being used to clear debris). The “correlated noise” signal represents just the chainsaw’s signal, as detected by a second monaural microphone located on the firefighter’s waist belt or lower torso, but nowhere near his/her mouth. The adaptive filter’s purpose is to enhance the voice signal so that the firefighter may communicate effectively, for example, when using a radio.

Since the chainsaw signal (the “noise”) travels a slightly different path on its way to the two inputs shown in Fig. 3, it is correlated but not equal in those two inputs. Thus, subtraction alone is not sufficient to mitigate the noise; the FIR filter adapts to model the effects of the path difference between the two microphones and better cancel out the noise. As the firefighter moves, or the chainsaw user moves (or starts and stops the chainsaw cutting), the difference also changes. The filter coefficients will adapt and be changed on the next update to best cancel out the noise.

For this demonstration, we chose to implement an LMS algorithm to update the FIR coefficients of the adaptive filter. Note that since two monaural microphones are connected to the single stereo “microphone in” jack of the LCDK, a simple splitter cable must be fabricated or purchased.

## Classroom Results

During the Fall 2012 semester at Boise State University, three weeks of our traditional DSP course was devoted to teaching digital filter design. Since a thorough coverage of filter design concepts can easily span an entire academic year, and adaptive filters in particular are routinely taught as a sep-

arate course, we had a dilemma: we wanted to expose the students to adaptive filters, but didn't want to present the topic in such an abbreviated way that they would quickly forget it. After some thought, we decided to go ahead and devote the single class period we could carve out for adaptive filters, to help whet our student's appetite for additional DSP content. This additional content could then come (in later semesters) in the form of additional coursework, senior design projects, follow-on graduate studies, or research.

Given the severe time constraint of a single class period, only the noise canceling configuration of adaptive filters was discussed. The fact that an almost unlimited number of adaptive structures exist was also mentioned to motivate additional student interest.

Following our established three-step pedagogy mentioned earlier, we first reviewed the underlying theory. After presenting the basic idea and the block diagram shown in Fig. 3, we used winDSK8 to show the time and frequency domain signals from two monaural microphones (using the Oscope/Analyzer function from the Filters/Communications block) to show how difficult it would be to manually try to determine the characteristics of the needed FIR filter in Fig. 3. We then conducted non-real-time MATLAB simulations (step two), showed the students what the "de-vectorized" MATLAB code looked like, and then ported the algorithm into C for execution in real-time on a LCDK DSP board (step three). The final real-time demonstration, in which the noise was essentially canceled, was met with considerable enthusiasm by the students.

At the end of the semester, student opinions of various topics in the course were measured using a five-point Likert-scale survey. The survey item most pertinent to this paper was,

*"The adaptive filter demonstration helped me understand the underlying concepts."*

The allowed responses were,

1. strongly disagree
2. disagree
3. undecided
4. agree
5. strongly agree.

The average response of the 14 survey participants to this statement was 3.86, with a standard deviation of 1.03. Of the 14, only two students circled the "2 - disagree" response and none circled the "1 - strongly disagree" response. See Figure 4 for the complete results. Note that this small sample size is inadequate to draw any statistically firm conclusions, but we still are confident in the general results.

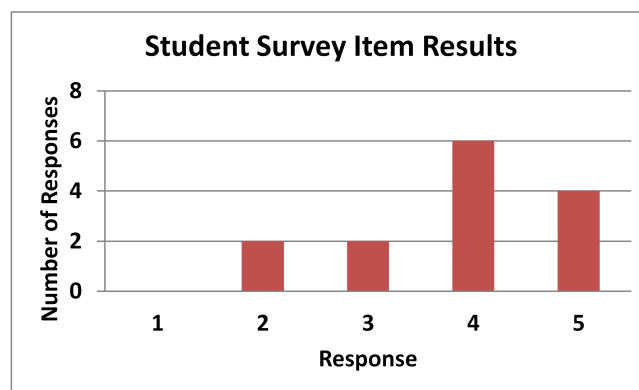


Figure 4: Raw data for student responses to the item, "The adaptive filter demonstration helped me understand the underlying concepts."

## Conclusions

The majority of our students definitely agreed with our belief that the adaptive filter demonstration helped them better understand the underlying concepts. This was in spite of the fact that only one class period was spent on the topic. A similar survey question asking about the efficacy of DSP demonstrations in general averaged a score of 4.29 on the same scale. This certainly is in line with our anecdotal information regarding the teaching effectiveness of demonstrations in the classroom. Ideally, we would like to find a valid way to measure

retention of the topic in these students a semester or two later.

We strongly encourage faculty who teach DSP to incorporate demonstrations and hands-on experience with real-time hardware for their students. We have made various resources widely available (many of which are free) to help in this endeavor [26,27].

### References

1. C. H. G. Wright, T. B. Welch, D. M. Etter, and M. G. Morrow, "Teaching DSP: Bridging the gap from theory to real-time hardware," *ASEE Comput. Educ. J.*, pp. 14–26, July–September 2003.
2. C. H. G. Wright, M. G. Morrow, M. C. Allie, and T. B. Welch, "Using real-time DSP to enhance student retention and engineering outreach efforts," *ASEE Comput. Educ. J.*, pp. 64–73, October–December 2008.
3. C. S. Burrus, "Teaching filter design using MATLAB," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 20–30, Apr. 1993.
4. R. F. Kubichek, "Using MATLAB in a speech and signal processing class," in *Proceedings of the 1994 ASEE Annual Conference*, pp. 1207–1210, June 1994.
5. R. G. Jacquot, J. C. Hamann, J. W. Pierre, and R. F. Kubichek, "Teaching digital filter design using symbolic and numeric features of MATLAB," *ASEE Comput. Educ. J.*, pp. 8–11, January–March 1997.
6. J. H. McClellan, C. S. Burrus, A. V. Oppenheim, T. W. Parks, R. W. Schafer, and S. W. Schuessler, *Computer-Based Exercises for Signal Processing Using MATLAB 5*. MATLAB Curriculum Series, Upper Saddle River, NJ (USA): Prentice Hall, 1998.
7. J. W. Pierre, R. F. Kubichek, and J. C. Hamann, "Reinforcing the understanding of signal processing concepts using audio exercises," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 6, pp. 3577–3580, Mar. 1999.
8. C. H. G. Wright and T. B. Welch, "Teaching DSP concepts using MATLAB and the TMS320C31 DSK," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 6, pp. 3573–3576, Mar. 1999.
9. M. G. Morrow and T. B. Welch, "winDSK: A windows-based DSP demonstration and debugging program," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 6, pp. 3510–3513, June 2000. (invited).
10. M. G. Morrow, T. B. Welch, C. H. G. Wright, and G. W. P. York, "Demonstration platform for real-time beamforming," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 2693–2696, May 2001.
11. C. H. G. Wright, T. B. Welch, D. M. Etter, and M. G. Morrow, "Teaching hardware-based DSP: Theory to practice," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 4148–4151, May 2002.
12. T. B. Welch, R. W. Ives, M. G. Morrow, and C. H. G. Wright, "Using DSP hardware to teach modem design and analysis techniques," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. III, pp. 769–772, Apr. 2003.
13. T. B. Welch, M. G. Morrow, and C. H. G. Wright, "Using DSP hardware to control your world," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. V, pp. 1041–1044, May 2004. Paper 1146.
14. T. B. Welch, C. H. G. Wright, and M. G. Morrow, "Caller ID: An opportunity to teach DSP-based demodulation," in *Proceedings of*

- the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. V, pp. 569–572, Mar. 2005. Paper 2887.
15. T. B. Welch, C. H. G. Wright, and M. G. Morrow, “Teaching rate conversion using hardware-based DSP,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. III, pp. 717–720, Apr. 2007.
  16. C. H. G. Wright, M. G. Morrow, M. C. Allie, and T. B. Welch, “Enhancing engineering education and outreach using real-time DSP,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. III, Apr. 2008.
  17. T. B. Welch, C. H. G. Wright, and M. G. Morrow, “Software defined radio: inexpensive hardware and software tools,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 2934–2937, Mar. 2010.
  18. M. G. Morrow, C. H. G. Wright, and T. B. Welch, “winDSK8: A user interface for the OMAP-L138 DSP board,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 2884–2887, May 2011.
  19. M. G. Morrow, C. H. G. Wright, and T. B. Welch, “Old tricks for a new dog: An innovative software tool for teaching real-time DSP on a new hardware platform,” *ASEE Comput. Educ. J.*, pp. 64–69, October–December 2011.
  20. T. B. Welch, C. H. G. Wright, and M. G. Morrow, *Real-Time Digital Signal Processing: From MATLAB to C with C6x DSPs*. Boca Raton, FL (USA): CRC Press, 2nd ed., 2012.
  21. Texas Instruments, “L138/C6748 Development Kit (LCDK),” 2012. [http://processors.wiki.ti.com/index.php/L138/C6748\\_Development\\_Kit\\_\(LCDK\)](http://processors.wiki.ti.com/index.php/L138/C6748_Development_Kit_(LCDK)).
  22. S. D. Stearns, *Digital Signal Processing with Examples in MATLAB*. Boca Raton, FL (USA): CRC Press, 2003.
  23. A. D. Poularikas and Z. M. Ramadan, *Adaptive Filtering Primer with MATLAB*. Boca Raton, FL (USA): CRC Press, 2006.
  24. S. Haykin, *Adaptive Filter Theory*. Upper Saddle River, NJ (USA): Prentice Hall, 1996.
  25. R. M. Rangayyan, *Biomedical Signal Analysis: A Case-Study Approach*. New York: John Wiley & Sons, 2001.
  26. “RT-DSP website.” <http://www.rt-dsp.com>.
  27. Educational DSP (eDSP), L.L.C., “DSP resources for TI DSKs,” 2012. <http://www.educationaldsp.com/>.

### **Biographical Information**

Michael G. Morrow, P.E., is with the Department of Electrical and Computer Engineering at the University of Wisconsin, Madison, WI. His research interests include real-time digital systems, embedded system design, software engineering, curriculum design, and educational assessment techniques. He is a member of ASEE and IEEE. E-mail: [morrow@ieee.org](mailto:morrow@ieee.org)

Cameron H. G. Wright, Ph.D, P.E., is with the Department of Electrical and Computer Engineering at the University of Wyoming, Laramie, WY. His research interests include signal and image processing, real-time embedded computer systems, biomedical instrumentation, and engineering education. He is a member of ASEE, IEEE, SPIE, BMES, NSPE, Tau Beta Pi, and Eta Kappa Nu. E-mail: [c.h.g.wright@ieee.org](mailto:c.h.g.wright@ieee.org)

Thad B. Welch, Ph.D, P.E., is with the Department of Electrical and Computer Engineering at Boise State University, ID. His research interests include implementation of communication systems using DSP techniques, DSP education, multicarrier communication systems analysis, and RF signal propagation. He is a member of ASEE, IEEE, Tau Beta Pi, and Eta Kappa Nu. E-mail: [t.b.welch@ieee.org](mailto:t.b.welch@ieee.org)