COMMFSK: A HARDWARE APPROACH TO TEACHING FSK

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

Michael G. Morrow
Department of Electrical and Computer Engineering
University of Wisconsin-Madison, WI

Gerald Vineyard
Department of Electrical and Computer Engineering
U.S. Naval Academy, MD

Abstract

Many communication topics are difficult for undergraduate students to internalize, but demonstrations and laboratory experiences have been shown to help a great deal. This paper describes how we teach the concept of frequency shift keying by using a highly successful combination of theory, demonstrations, lab exercises, and real-time DSP experiences that incorporate MATLAB and the Texas Instruments C67x digital signal processing starter kit.

Introduction

While many communication topics are difficult for undergraduate students to fully understand, the use of demonstrations and laboratory experiences have been shown to greatly facilitate the learning process [1–7]. This paper describes how to teach the digital communication modulation technique of frequency shift keying (FSK) using a highly successful combination of theory, demonstrations, lab exercises, and real-time DSP experiences that incorporates MATLAB® and the Texas Instruments (TI) C67x digital signal processing starter kit (DSK). This approach, when combined with the appropriate test and measurement equipment, provides a superior method of reinforcing communications theory and many of the associated practical, real-world tradeoffs that students often overlook.

Specifically, this paper describes the addition of a frequency shift-keying (FSK) capability to the winDSK6 program [6]. FSK is one of the most straight-forward forms of digital communication. Once students understand FSK it is far easier to help them understand more complicated methods of digital modulation [8–10]. This FSK capability is incorporated in a new winDSK6 module, called CommFSK, which includes the following features:

- generation of phase continuous and phase discontinuous FSK with adjustable data rate;
- control of the modulated signal's amplitude, center frequency, and frequency deviation;
- source data selection from a pattern of alternating 0's and 1's, several PN-sequences, random data, all 0's, all 1's, ASCII text messages from keyboard, or data from files;
- optional built-in or user-defined asynchronous communications protocol;
- user defined FIR-based filtering of the resulting FSK signal; and,
- full integration into the winDSK6 program.

Why winDSK6?

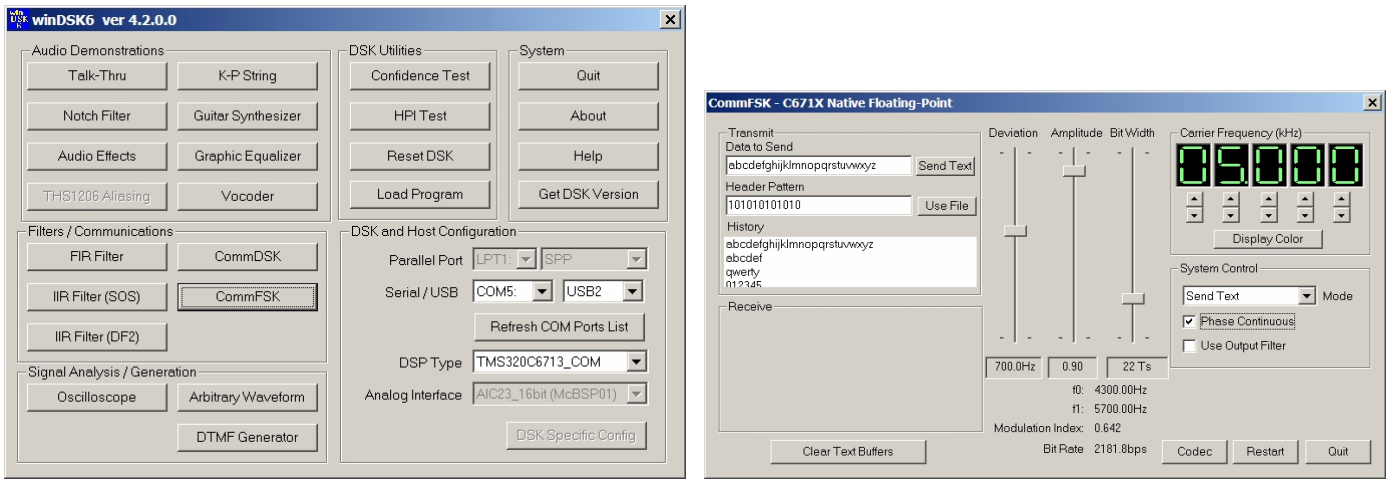
The winDSK6 program is a Microsoft Windows® application control program for a TI TMS320C6x DSK [11]. The C6x DSK includes a version of TI's Code Composer Studio (CCS), a professional level integrated development environment for DSP code generation, debugging, and analysis. While this greatly enhances the ability of skilled programmers to implement significant DSP algorithms in a laboratory setting, using CCS makes it more difficult to perform classroom demonstrations for several reasons. First, by using CCS the DSK hardware is dependent upon CCS driver files, meaning the DSK hardware could only be used on machines with CCS installed. This can pose a serious burden for educators who want to perform demonstrations in classrooms or laboratories where CCS is not installed, due to a lack of sufficient licenses or due to not having time to install and uninstall CCS for a demonstration. A related problem we found is that driver modifications made by TI as part of updates to CCS sometimes cause compatibility problems with previously working demos. Another issue is related to the extensive professional-grade capabilities of CCS itself: it is a very complex software package, and to the novice or student it is often overwhelming.

For example, if your pedagogical goal is to get students to experiment with various digital modulation schemes implemented on a DSK and observe the results in real-time, then using CCS for this purpose may be overkill and a potential impediment to learning. In order to take students on a journey from theory to real-time practice, there needs to be an infrastructure in place to support them and target as many modes of learning as are reasonably possible [12]. One very important segment of this needed infrastructure is a tool to support faculty demonstrations, student experimentation, and student self-learning. Professional-grade tools such as CCS may, at first, be too complex to lend themselves easily to this pursuit.

One solution designed to overcome these difficulties is winDSK6 [13]. The winDSK6 program is a Windows application that provides an intuitive and easy-to-use interface, and ensures that a student's first experience with the DSK is a positive and motivating one. It makes the DSK hardware much more accessible to students, and facilitates easy-to-use, ready-made classroom and laboratory demonstrations. For simplicity, all application software and DSK code is embedded in the executable file; this eliminates the requirement to have CCS installed on the machine. A help file provides a section on each demonstration that discusses the theory and operation of the application, and context-specific help is available on each application control. The winDSK6 program fully encapsulates the DSK's physical and logical interface to the host computer.

The applications that form the basis of winDSK6 have evolved over time and experience, with the authors' needs in the classroom and laboratory being the motivating force behind the new capabilities within each new version. Individual applications are all dialog-based, and perform a similar sequence of operations to execute an application program. This all occurs upon selecting an application with a mouse click; to the user it seems to occur instantaneously. Once the application is running, communication via the shared memory block is used to control the DSK application's behavior in response to user input via the dialog window displayed on the host computer. This gives users real-time interactivity and immediate feedback when changes are made on the host computer.

As shown in Figure 1(a), the currently available applications in winDSK6 include various audio demonstrations, digital filters/communication system examples, signal analysis and generation (including an oscilloscope/spectrum analyzer capability), and highly useful utilities to help debug possible hardware problems. Selecting the CommFSK option from the winDSK6 user interface launches the CommFSK application and brings up the window shown in Figure 1(b). The remainder of this paper will discuss the features of the new FSK digital communication transmitter and receiver provided by CommFSK.



(a) (b)

Figure 1: (a) The main winDSK6 user interface. (b) The CommFSK user interface. Here the center (carrier) frequency is set to 5 kHz, frequency deviation Δf from center is 700 Hz, the data rate 2181.8 bps, the type of FSK is phase continuous, the data source is text (with the FSK receiver not active), a user defined 12-bit synchronization header of 101010101010 is being used, and no filtering is being performed on the modulated output.

CommFSK Features

The capabilities of CommFSK were listed in the introduction. In particular, it can be seen from Figure 1(b) that various parameters can be easily adjusted while the program is running in real-time on the C67x DSK.

Some parameters, such as modulation index h, are not set directly but only as a result of setting other parameters more specific to FSK. For example, the basic definition of the modulation index for FSK is $h = 2\Delta f/r$, where Δf is the frequency deviation from the center (carrier) frequency, and r is the data rate in bps. Thus for the settings shown in Figure 1(b), frequency deviation $\Delta f = 700$ Hz, and the data rate $r = 2181.8$ bps, yielding a modulation index of $h = 0.642$.

While frequency deviation can be directly adjusted in the CommFSK application via a slider bar (called "Deviation"), the data rate is one of the indirectly controllable parameters. Since the sample rate for the DSK running in real-time normally remains fixed (if the user wishes to adjust any other CommFSK parameters), the data rate is adjusted by selecting, via another slider bar, the number of samples per bit (called "Bit Width"). Thus for the settings shown in Figure 1(b), the sample frequency for the DSK is 48 kHz and the number of samples per bit is 22, which yields a data rate of $(48\times10^3)/22 = 2181.8$ bps. If desired, the "Codec" button allows the user to change the sample frequency.

The third slider bar, "Amplitude," adjusts the output level of the FSK signal. Another adjustment to the output is provided via the "Use Output Filter" check box. Upon checking this box, another dialog box opens allowing the user to enter FIR filter coefficients either manually, by cut and paste, or from a file. Just above the "Use Output Filter" check box is the "Phase Continuous" check box; if the user un-checks this box the transition between the two FSK frequencies will not be phase continuous.

The most versatile user interface control for CommFSK is the drop-down list, called "Mode," that selects the type of data that is to be sent by the transmitter. As previously inferred, this control allows the user to select source data. The choices include: a pattern of alternating 0's and 1's, a PN-sequence, random data, all 0's, all 1's, ASCII text messages from the keyboard, or a file data. For the

text message modes, the user can choose whether or not the CommFSK receiver will be used to recover the data using a simple asynchronous communications protocol. The user can also specify an arbitrary protocol synchronization header of up to 256 bits, entered either manually or loaded from a text file.

Note that CommFSK does *not* have an option for implementing bit-shaping filters. Their use causes the input to the transmitter to be no longer binary, which requires the transmitter design to be slightly more complex, and we chose not to implement such a feature at this time.

Using CommFSK for Teaching

The CommFSK application running in real-time on the C67x DSK permits students to immediately observe the effects of changing the various parameters for FSK modulation. From this they can easily experiment, learn about, and develop practical insight about many aspects of this simple digital communications system.

One parameter of great interest is h, the modulation index [14]. By adjusting the data rate (by specifying the number of samples per bit via the "Bit Width" slider bar), and/or the frequency deviation, the resulting modulation index can be easily controlled. For example, Frerking [14] discusses the effect of h on the bandwidth of the FSK signal as shown in Figure 2; this can be verified using CommFSK and a spectrum analyzer as shown in Figure 3 for the case of $h = 0.64$. The spectrum analysis can be performed a variety of ways, such as using dedicated test equipment, using a PC-based tool such as LabVIEW, or even by using another DSK running the spectrum analyzer mode of winDSK6.

The previous examples showed phase continuous FSK. With CommFSK, students can immediately see (and/or hear, if the output is connected to speakers) how phase discontinuous FSK results in a very different output. Figure 4(b) and (c) show

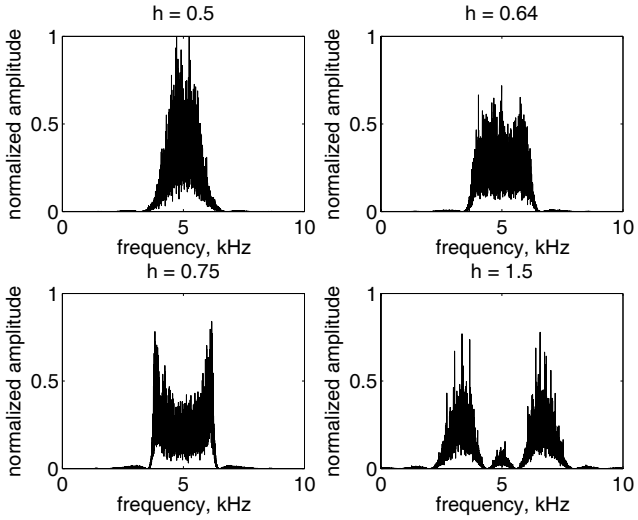


Figure 2: Examples of four modulation indices (h) for FSK. The center frequency is $f_c = 5$ kHz and the data rate is $r = 2400$ bps. Frequency deviation Δf was varied to change h.

an example of phase discontinuous FSK, where it should be obvious to the student that the transmitter effectively "jumps" back and forth between two oscillators to represent the 1's and 0's of the data.

A related topic is the effect of using a bandpass filter on the FSK output to constrain the required bandwidth. For example, Frerking [14] mentions that when $h = 1.5$, the required bandwidth is approximately $2r$, or twice the bit rate, and shows a

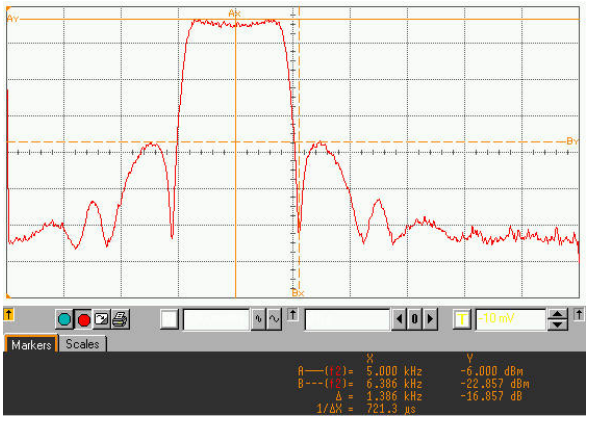
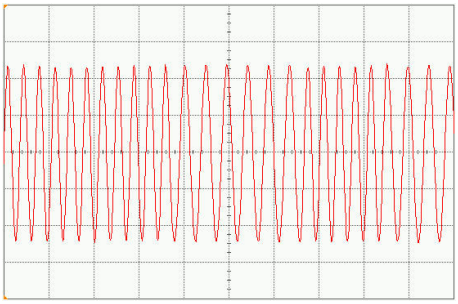
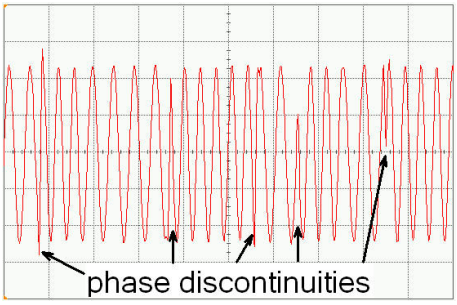


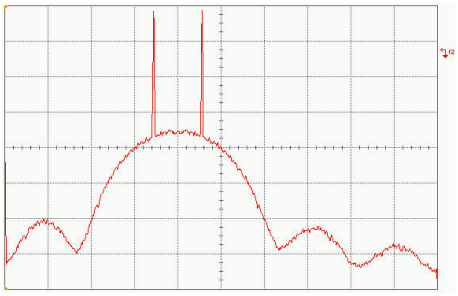
Figure 3: CommFSK output from a C6713 DSK displayed on a spectrum analyzer. The center frequency was $f_c = 5$ kHz and the actual modulation index was $h = 0.642$.



(a)



(b)



(c)

Figure 4: CommFSK output from a C6713 DSK displayed on an Agilent Infiniium 54810A oscilloscope/spectrum analyzer. Note: all modulation parameters are identical with those used to create Figure 3, except that (b) and (c) show phase discontinuous FSK. (a) Time domain, phase continuous FSK. (b) Time domain, phase discontinuous FSK. (c) Frequency domain, phase discontinuous FSK.

figure to this effect using a linear amplitude scale (similar to Figure 2). When viewed with amplitude in dB, the student can see that there are sidelobes that are only about 15 dB down that exceed the $2r$ bandwidth. The CommFSK application allows the student to specify an FIR filter (up to order 127) that will be applied to the FSK output. Upon selecting the check box to apply a filter, the dialog

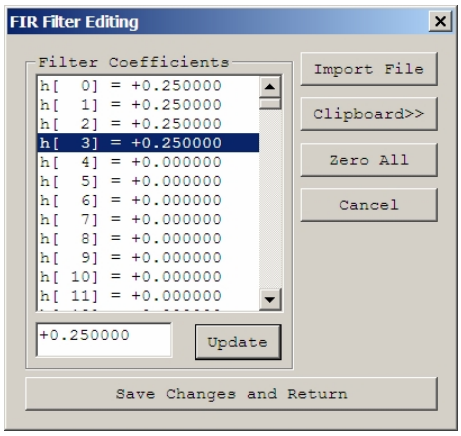


Figure 5: Dialog that pops up in commDSK when the output filter is selected. This permits easy entering of FIR filter coefficients manually, via the clipboard or from a file.

box shown in Figure 5 appears, allowing the student to enter FIR filter coefficients in a variety of ways.

In Figure 6, the left graph is the unfiltered phase continuous FSK output. The right graph shows the effect of applying to the FSK output a 71-order, linear phase equiripple (Parks-McClellan) FIR bandpass filter where the passband is from 2800–7200 Hz, the transition bandwidth at both ends of the passband is 1 kHz, the attenuation in the stopband is 40 dB, and the ripple in the passband is 1 dB. The filter was designed using MATLAB's sptool from the Signal Processing Toolbox. Students can use such an example to reinforce FIR filter design, use CommFSK to easily generate such a filtered FSK signal, and then investigate how the application of the filter may effect data recovery.

Another useful feature of CommFSK is the ability to send all 0's or all 1's (to check the actual tone frequencies, frequency stability, etc.), alternating 0's and 1's, random bits (using the C standard library `rand()` function), or PN sequences. For PN sequences, length 3, 5, and 7 shift register generation of maximal length sequences are supported. These can be useful, for example, if you have students design their own FSK receiver and correlate the recovered data with the known PN sequence and observe the results.

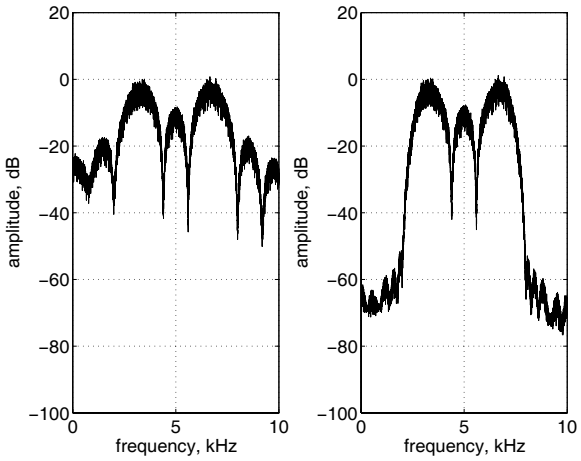


Figure 6: Two examples of FSK modulation with ($h = 1.5$). In both cases, the center frequency is $f_c = 5$ kHz, the data rate is $r = 2400$ bps, and the frequency deviation is $\Delta f = 1800$ Hz. Left: no filtering. Right: a 71-order, equiripple (Parks-McClellan) FIR bandpass filter was used on the output.

As mentioned previously, CommFSK includes not only a highly versatile transmitter but also a simple receiver. The receiver is used only for text data, using a simple asynchronous protocol (start bit, stop bit) similar to RS-232. The user can type in text, or load a text file, and see that the recovered data matches the original data. This allows one DSK to loop back the data, or two DSKs to "talk" to each other using FSK (the latter may appeal to the text messaging "addicts" among your students). But such a simple protocol is only adequate for low data rates, and we purposely designed CommFSK so that the user can increase the data rate to the point where the simple receiver fails. This provides to our students the "opportunity" to design their own binary synchronization header for an asynchronous communications protocol, using up to 256 bits. They can manually type in the desired pattern of 1's and 0's in a CommFSK text window, or store the pattern in an ASCII file and load it to CommFSK from there to save them from the tedium and the error-prone nature of retyping long bit sequences.

This section only scratches the surface of the many demonstrations and student exercises that can be performed using CommFSK. Educators are encouraged to e-mail the authors with a description of how they use this tool to help teach digital communications.

Conclusions

We have developed and described a significant enhancement to the winDSK6 program that provides an intuitive method to teach FSK digital communication signal generation and recovery both affordably and conveniently. The adjustable parameters in CommFSK are summarized in Figure 7, where the three main logical blocks of data source, FSK modulator, and output control are shown. The transmitter and receiver topics can be taught together or separately, depending upon the instructor's pedagogical goals. The hardware investment required to implement such an FSK communication system is rather modest, and all of the needed software has already been developed by the authors. With such tools available, our students are now motivated to design, build, and test not only their own FSK transmitters, but their own FSK receivers as well.

Once they are comfortable with FSK, we have found they are far more ready to understand other digital communication topics. Note that CommDSK, also a part of winDSK6, allows students to experiment in real-time with BPSK, QPSK, 8-PSK, 16-PSK, 8-QAM, and 16-QAM along with

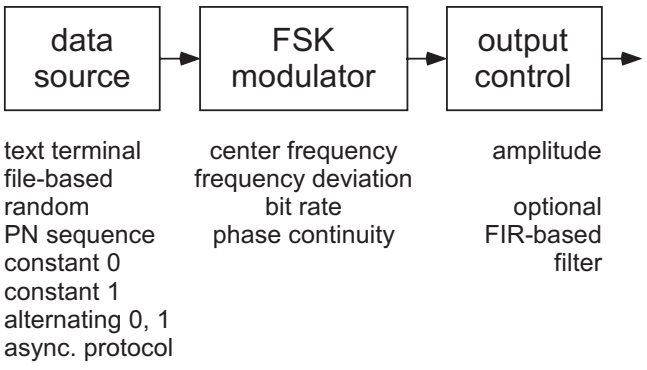


Figure 7: User-controlled aspects of the CommFSK transmitter are listed under the associated blocks.

topics such as bit shaping, I/Q imbalance, channel noise, etc. [15].

We freely distribute the winDSK6 software for educational, non-profit use, and invite user suggestions for improvement. See http://eceserv0.ece.wisc.edu/ morrow/software/. Interested parties are also invited to contact the authors via e-mail.

References

1. T. B. Welch, C. H. G. Wright, and M. G. Morrow, *Real-Time Digital Signal Processing: From MATLAB to C with the TMS320C6x DSK*. CRC Press, 2006.

2. ——, "Caller ID: An opportunity to teach DSP-based demodulation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. V, Mar. 2005, pp. 569–572, paper 2887.

3. ——, "Experiences in offering a DSP-based communication laboratory," in *Proceedings of the 11th IEEE Digital Signal Processing Workshop and the 3rd IEEE Signal Processing Education Workshop*, Aug. 2004.

4. 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, Apr. 2003, pp. 769–772.

5. M. G. Morrow, T. B. Welch, and C. H. G. Wright, "A tool for real-time DSP demonstration and experimentation," in *Proceedings of the 10th IEEE Digital Signal Processing Workshop*, Oct. 2002, paper 4.8.

6. 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, June 2000, pp. 3510–3513, (invited).

7. M. G. Morrow, T. B. Welch, and C. H. G. Wright, "An inexpensive software tool for teaching real-time DSP," in *Proceedings of the 1st IEEE DSP in Education Workshop*. IEEE Signal Processing Society, Oct. 2000.

8. L. W. Couch, II, *Digital and Analog Communication Systems*, 6th ed. Prentice Hall, 2001.

9. B. Sklar, *Digital Communications: Fundamentals and Applications*, 2nd ed. Prentice Hall, 2001.

10. C. H. G. Wright, T. B. Welch, and M. G. Morrow, "An inexpensive method to teach hands-on digital communications," in *Proceedings of the IEEE/ASEE Frontiers in Education Annual Conference*, Nov. 2003, pp. F2E19–F2E24.

11. M. G. Morrow, T. B. Welch, and C. H. G. Wright, "An introduction to hardware-based DSP using winDSK6," in *Proceedings of the 2001 ASEE Annual Conference*, June 2001, session 1320.

12. 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," in *Proceedings of the 2002 ASEE Annual Conference*, June 2002.

13. T. B. Welch, D. M. Etter, C. H. G. Wright, M. G. Morrow, and G. J. Twohig, "Experiencing DSP hardware prior to a DSP course," in *Proceedings of the 10th IEEE Digital Signal Processing Workshop*, Oct. 2002, paper 8.5.

14. M. E. Frerking, *Digital Signal Processing in Communication Systems*. Van Nostrand Reinhold, 1994, 7th printing by Kluwer Academic Publishers, 2000.

15. T. B. Welch, M. G. Morrow, C. H. G. Wright, and R. W. Ives, "commDSK: a tool for teaching modem design and analysis," *ASEE Comput. Educ. J.*, vol. XIV, no. 2, pp. 82–89, Apr. 2004.

Biographical Information

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

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

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, DSP education, and software engineering. He is a member of ASEE and IEEE. E-mail: morrow@ieee.org

Gerald Vineyard is a Midshipman in the Department of Electrical and Computer Engineering at the U. S. Naval Academy in Annapolis, MD. Mr. Vineyard's academic interest include real-time DSP, power electronic system design, and pulsed power sources for electromagnetic launching systems. E-mail: geraldvineyard@yahoo.com