REAL WORLD ULTRASONIC SIGNALS AND THEIR APPLICATION IN TEACHING SIGNAL PROCESSING

Thad B. Welch
Department of Electrical and Computer Engineering
Boise State University, ID

Cameron H. G. Wright
Department of Electrical and Computer Engineering
University of Wyoming, WY

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

Abstract

In our never-ending quest to find ways to interest and motivate our students, we have recently found something new for our "bag of teaching tricks." Ultrasonic signals present a unique andragogical opportunity in any course where signal processing theory and techniques are taught. The authors have recorded (or obtained) a number of naturally occurring ultrasonic signals (e.g., bat echolocation sounds and dolphin whistles) as well as artificially generated ultrasonic signals (e.g., output from a dog whistle and signals from a device from ThinkGeek called an Annoy-a-tron). This paper discusses how these signals can be effectively used to teach, demonstrate, and reinforce the signal processing concepts of time dilation/compression, frequency translation, spectral analysis/estimation, and aliasing.

Introduction

Keeping our students interested and motivated is an ongoing challenge for professors today. In the authors' experience, we find that our students of signal processing still respond well to demonstrations that pique their curiosity or interest them in one way or another. This usually requires involving their sense of hearing and/or sight in some way. For most students, tying such demonstrations to real world applications can bring signal processing theory to life [1–11]. Additionally, these applications, whether using real world signals, systems, or both, provide immediate relevance for what students may otherwise view as *just more theory*. Our students are quick to tell us that what they view as "theory for theory's sake" does not hold much interest for them.

Real-time digital signal processors, for example the popular C6x series from Texas Instruments, can easily generate algorithm-based signals and system implementations of devices such as filters, transmitters, receivers, and so forth [6, 12–15], but there is another category of interesting signals that instead need to be recorded prior to a demonstration. The recording of audible signals has become all but trivial using a modern a computer (with a built in sound card) and a microphone. Editing these recordings as needed using free (and in some cases open source) software, such as the excellent Audacity program [16], has allowed volunteers and others to create huge libraries of recorded sounds [17, 18]. This vast library of sounds, augmented by a few files that the authors have recently recorded, make up the basis of the signal processing lessons and demonstrations discussed in this paper.

One source of "ultrasonic" signals, the ThinkGeek Annoy-a-tron 2.0, [19] can produce sounds that are quite annoying, but only to people young enough to be able to hear them! Technically speaking, they are only "relatively ultrasonic" and are so only to older adults (since sensitivity to higher frequencies of the audio spectrum universally decreases with age). Thus, in a typical situation, the professor can blissfully generate various annoying sounds that can only be heard by the younger students in the room! This often generates animated discussions that range from electrical engineering topics such as frequency response and filtering to biomedical engineering topics such as age-related deterioration of hearing and the ramifications for hearing aids and other such devices.

Ultrasonic Signals Provide Effective Teaching Opportunities

Bats, for some reason, are very interesting to many students, and certain sounds created by bats are ultrasonic. While many bat sounds are available at the `soundbible.com` website [18], most of those signals were recorded with traditional audio recording equipment. Some universities (e.g., the University of Bristol in the United Kingdom) have used high speed digitizers sampling at up to 500,000 samples/second (500 ksps) to make their recordings. The signals that we used were recorded at these ultrasonic sample rates, then played back at the traditional compact disc (CD) sample rate (44.1 ksps) as explained below. These signals are available as WAV files, which makes their analysis using MATLAB quite simple. MATLAB can directly import a WAV file, while most of the other popular audio file formats are not as well supported. This is only a minor inconvenience, however, in that software such as the free Audacity program can easily convert between most popular audio file formats. We use MATLAB's ability to easily generate spectrograms, a time-frequency display where the horizontal axis is time and the vertical axis is frequency.

As an example of an ultrasonic signal many students find interesting, we import a "bat chirp" signal into MATLAB; we call the signal `data` and the sample frequency `fs`. We execute the command `specgramdemo(data,fs)`, generating the spectrogram shown in Figure 1. If the signal is stereo, your data variable will most likely have two columns. In this case, select a single column using a command such as `specgramdemo(data(:,1),fs)` to display just channel 1. The actual sample frequency of the recording shown in Figure 1 was 441 ksps, which is ten times the CD rate. The horizontal portion of the cross hair in Figure 1 was placed very close to 2 kHz. When the effect of the "times ten" time dilation (i.e., a 441 ksps signal played back at 44.1 ksps) is taken into account, this would represent the upper limit of "perfect" human hearing for a young person (approximately 20 kHz). Since almost all of the signal energy is "above" this line (at higher frequencies), it is fairly clear to most students why the signal is classified as ultrasonic. The whole concept of changing the sample frequency from recording to playback and the resulting time dilation effects shown in this demonstration using a signal such as a bat chirp presents a very nice teaching opportunity that keeps students quite engaged in the topic.

The sample frequency specified as the second argument of the `specgramdemo` command will appropriately scale the frequency (vertical) axis. However, note that using the audio playback button from the `specgramdemo` window may result in an error message, such as,

```
???  Error using ==>
audioplayer.audioplayer>audioplayer.resume
at 710 Device Error:  Invalid sample rate
```

if you have exceeded the capability of your computer's sound card.

Also visible in Figure 1 are the second and third harmonics of the bat's chirp. Zooming in on a similar bat chirp signal's harmonic structure results in Figure 2. This particular bat chirp was selected because several harmonics are easily visible. Remembering that the bat's chirp is always *decreasing* in frequency as time increases, what is causing

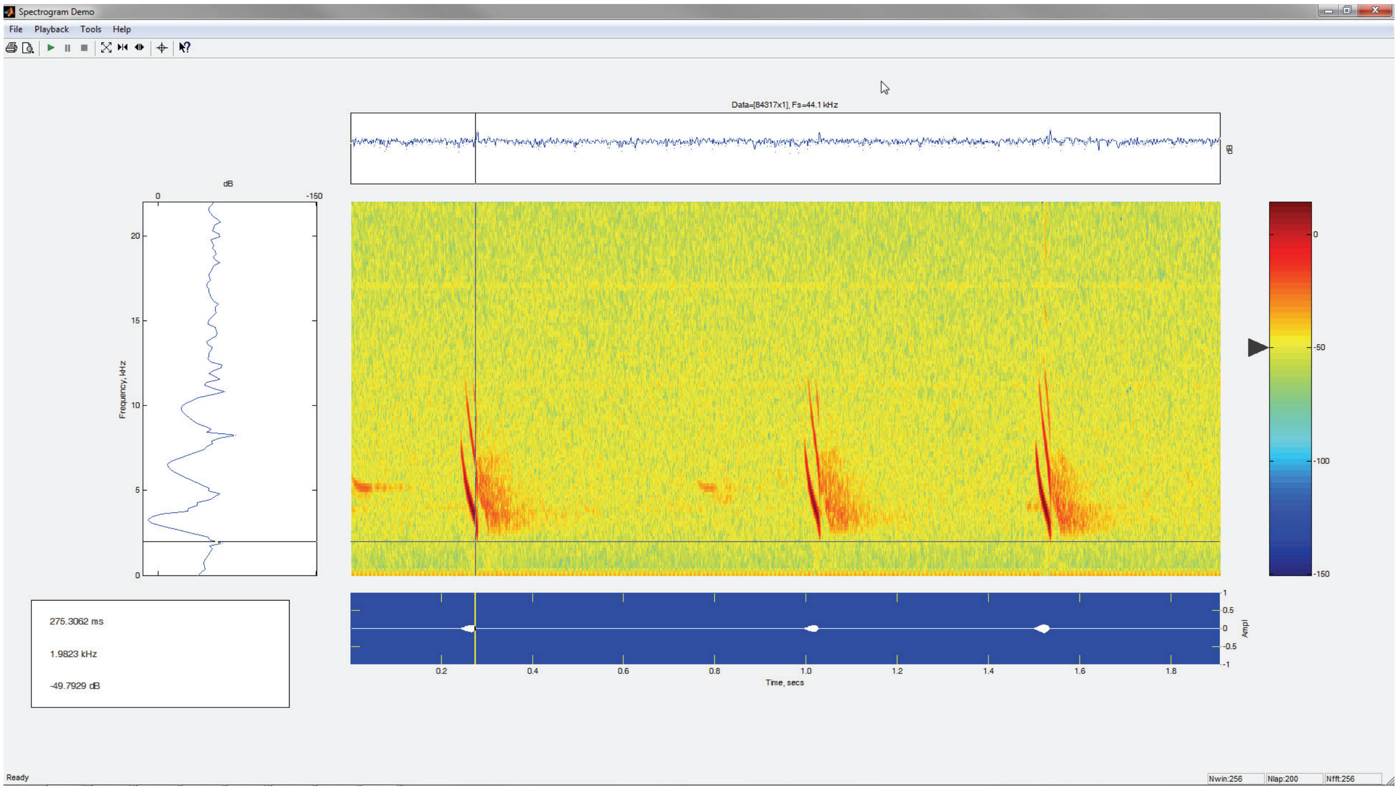

Figure 1: Spectrogram of a bat's chirp.

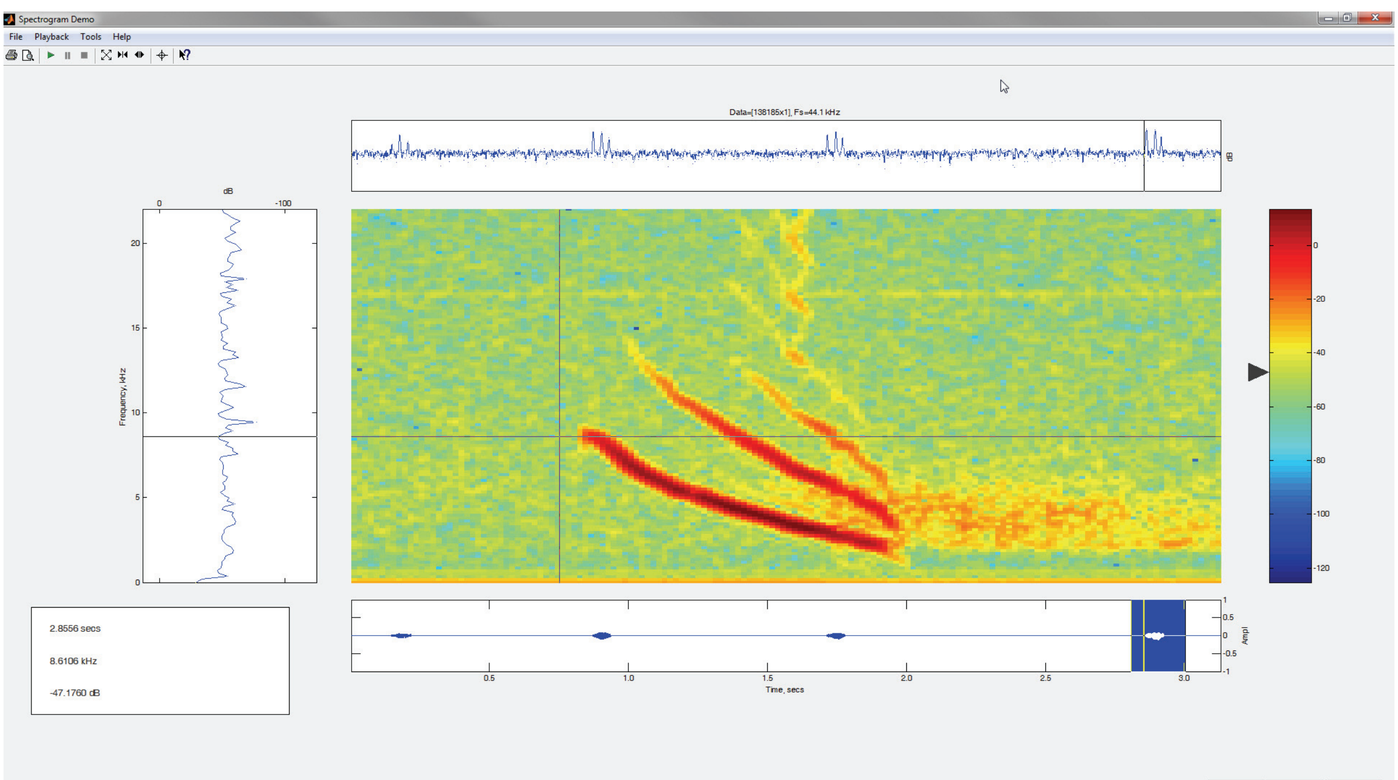

Figure 2: Spectrogram of a bat's chirp (zoomed in).

some portions of the signal to apparently *increase* in frequency? Aliasing is, at least in part, responsible for the increasing frequency portion of the signal. This can provide the professor with an almost effortless segue into the topic of aliasing, fold-over frequency, and related concepts.

Time dilation (i.e., signal expansion) can be achieved by lowering the sample frequency specified in the `specgramdemo` command. This will play the signal back more slowly, and all frequencies will be lowered. Alternatively, time contraction (i.e., signal compression) can be achieved by increasing the sample frequency in the `specgramdemo` command. This will play the signal back more quickly, and all frequencies will be increased. In both cases, the frequency axis of the spectrogram is automatically and appropriately rescaled and relabeled. Playing the signal and listening to the results while observing the plot of the spectrogram involves multiple senses and helps most students tremendously with their understanding of the associated theoretical concepts.

Similar discussions and demonstrations can be accomplished using signals emanating from dog whistles and from the two higher frequency signals that the Annoy-a-tron 2.0 generates. The term *higher* is intentionally used since these signals are not ultrasonic in the strict sense of the term, but are beyond the hearing of the authors (and most anyone older than 30)! This is where the Annoy-a-tron gets its name, in that it annoys people with *normal* hearing and leaves the older adults in the crowd unaffected. The reader can imagine many uses for such a device, we suspect.

Multirate Signal Processing

There are also situations where the signal being analyzed has been significantly oversampled. That is, the signal was sampled at a considerably higher frequency than twice the highest frequency present in the signal. This can be seen in Figure 3, where the vast majority of the dolphin's whistle energy is below 3 kHz. The effective sample rate for this signal is 44.1 ksps, so the "fold-over frequency" (the upper bound of the vertical axis shown in the figure) is 22.05 kHz, far above 3 kHz. Compare the original spectrogram of Figure 3 to the one shown in Figure 4, where the signal has been decimated by a factor of ten. The upper bound of the vertical axis shown in the figure (the "fold-over frequency") is now 2.2 kHz. Decimation by ten means that nine out of every ten samples is discarded. This effectively lowers the sample frequency by a factor of ten, since only one in ten samples is retained. This can be accomplished using the MATLAB command,

```
specgramdemo(downsample(data,10),fs/10).
```

The sampling requirements dictated by Nyquist must be consider *prior* to any decimation operation to prevent aliasing. Alternatively, if the signal is not bandlimited, you could lowpass filter it to *make* it bandlimited. Again, the effect of the decimation operation must be considered in the filter design to prevent aliasing. Once again, this provides an effortless segue into various important DSP topics. It should be noted that, in this properly designed demonstration, both versions of the signal sound the same!

The need to adjust the sample frequency and the ramifications associated with that could take up an entire course in a signal processing curriculum, but few professors have that luxury. In this example using a dolphin whistle, it's relatively easy to make it clear to students fairly quickly, as part of a broader course, that sampling far faster than is required by Nyquist has its drawbacks, and that there are well-established techniques for lowering the effective sample frequency. By using what many students view as an "interesting" signal, such as the dolphin whistle shown here, we are better able to maintain their interest in the topic, and find that in doing so they tend to retain the concept better.

The need to change the sample frequency of a signal comes up often in practical signal processing, but the proper technique is many times misunderstood. Whether the signal is played back faster

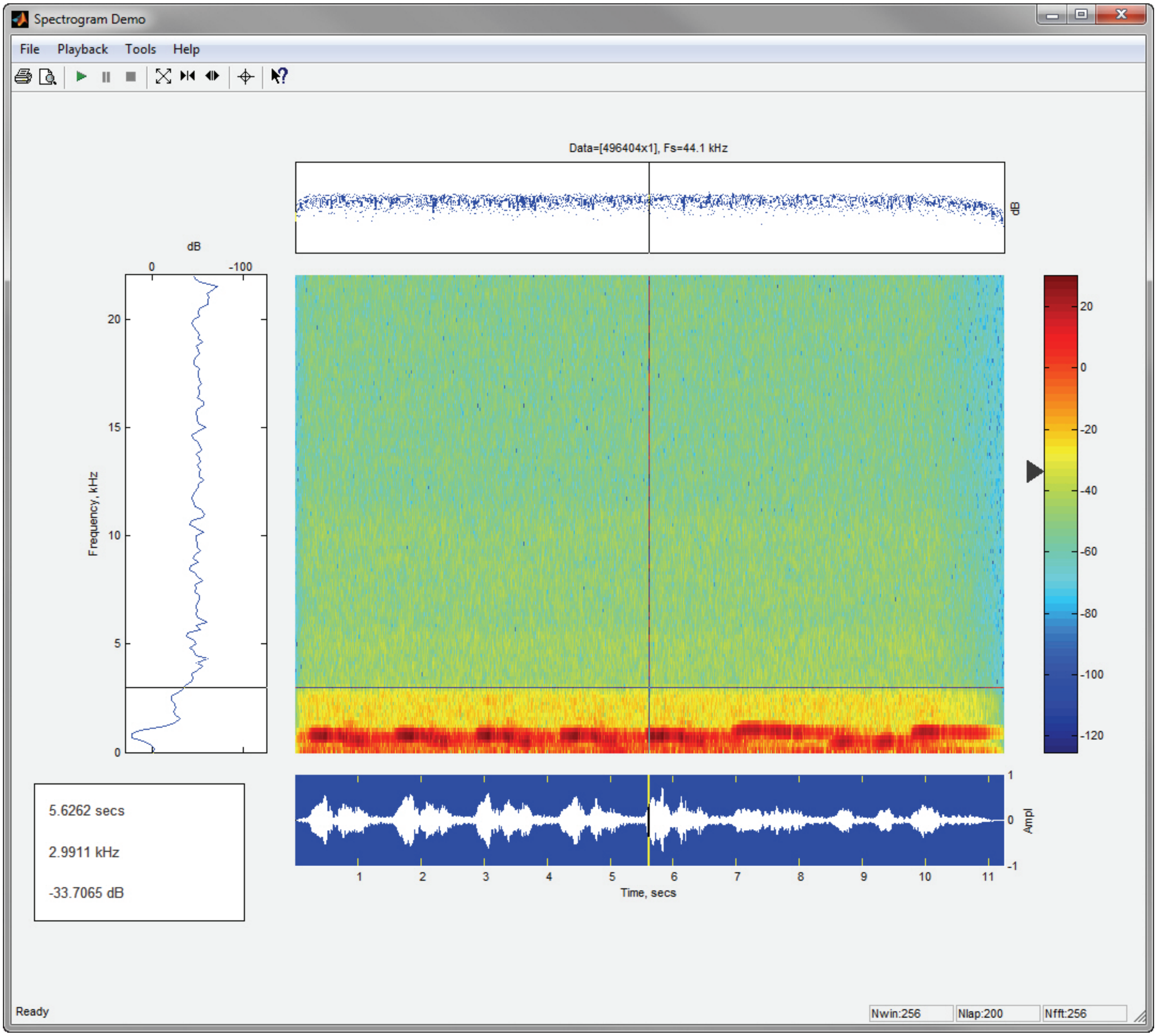

Figure 3: Spectrogram of a dolphin's whistle sampled at 44.1 ksps; the "fold-over frequency" is therefore $F_s/2 = 22.05$ kHz. This signal has been significantly oversampled.

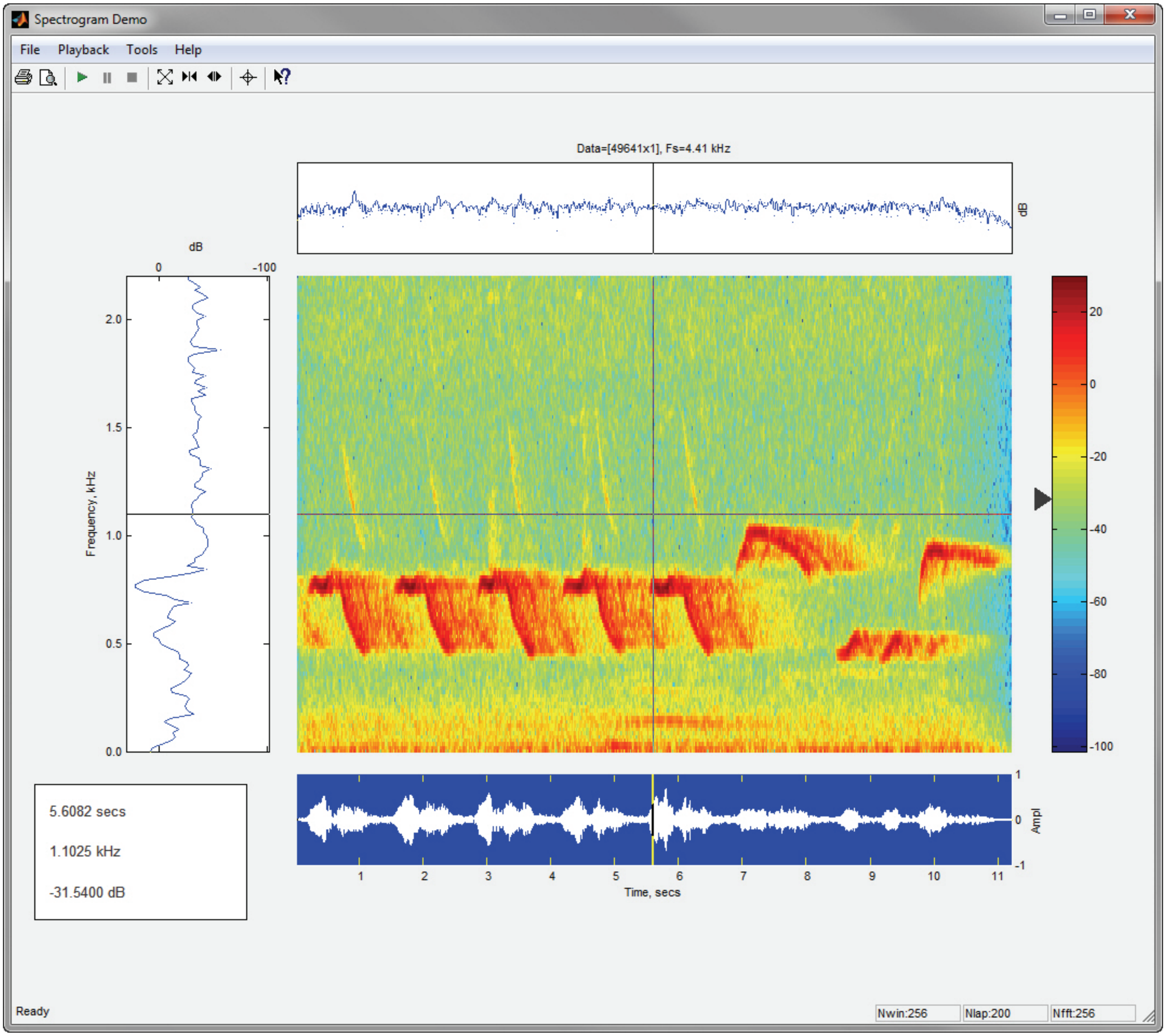

Figure 4: Spectrogram of a dolphin's whistle that has been decimated by a factor of ten, yielding an effective sample rate of 4.41 ksps and a "fold-over frequency" of 2.2 kHz.

or slower to aide in the audio interpretation of the signal, or to better understand the signal's spectral content, a number of real world examples using such "interesting" signals can help students truly understand the concept.

Conclusions

We have recently found that ultrasonic signals provide something new for our "bag of teaching tricks" to help interest and motivate our students. Naturally occurring ultrasonic signals (e.g., bat echolocation sounds and dolphin whistles) as well as artificially generated ultrasonic signals (e.g., output from a dog whistle and signals from a ThinkGeek Annoy-a-tron 2.0) all seem to interest our students more than most of the signals traditionally used in signal processing courses. This paper briefly described how these ultrasonic signals can be effectively used to teach, demonstrate, and reinforce signal processing concepts such as time dilation/compression, frequency translation, spectral analysis/estimation, and aliasing. We encourage other professors to give ultrasonic signals a try!

A number of these signals will be made available on the web site that supports the real-time DSP book, *Real-Time Digital Signal Processing: From MATLAB to C with C6x DSPs* [11]. Browse to the "Additional signals" link at `http://rt-dsp.com/2nd_ed/index.html`.

References

1. 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.

2. M. A. Yoder, J. H. McClellan, and R. W. Schafer, "Experiences in teaching DSP first in the ECE curriculum," in *Proceedings of the 1997 ASEE Annual Conference*, June 1997. Paper 1220-06.

3. 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, Prentice Hall, 1998.

4. G. W. P. York, C. H. G. Wright, M. G. Morrow, and T. B. Welch, "Teaching real-time sonar with the C6711 DSK and MATLAB," *ASEE Comput. Educ. J.*, pp. 79–87, July–September 2002.

5. T. B. Welch, C. H. G. Wright, and M. G. Morrow, "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.

6. 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.*, pp. 82–89, April–June 2004.

7. G. W. P. York, T. B. Welch, and C. H. G. Wright, "Teaching real-time ultrasonic imaging with a 4-channel sonar array, TI C6711 DSK and MATLAB," *ISA Biomed. Sci. Intrum.*, vol. 41, pp. 370–375, Apr. 2005.

8. T. B. Welch, C. H. G. Wright, and M. G. Morrow, "Caller ID: A project to reinforce an understanding of DSP-based demodulation," *ASEE Comput. Educ. J.*, pp. 2–7, October–December 2006.

9. C. H. G. Wright, T. B. Welch, M. G. Morrow, and G. Vinyard, "CommFSK: A hardware approach to teaching FSK," *ASEE Comput. Educ. J.*, pp. 38–45, April–June 2008.

10. 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.

11. T. B. Welch, C. H. G. Wright, and M. G. Morrow, *Real-Time Digital Signal Processing: From MATLAB to C with C6x DSPs*. CRC Press, 2nd ed., 2012.

12. 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).

13. 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.

14. M. G. Morrow, C. H. G. Wright, and T. B. Welch, "WinDSK8: the continuing saga of winDSK," in *Proceedings of the 14th IEEE Digital Signal Processing Workshop*, Jan. 2011.

15. 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.

16. "Audacity: Free audio editor and recorder." `http://audacity.sourceforge.net/`.

17. "Pdsounds: free sound library." `http://www.pdsounds.org/`.

18. "Soundbible: the sound encyclopedia." `http://soundbible.com/`.

19. "The ThinkGeek Annoy-a-tron 2.0." `http://www.thinkgeek.com/gadgets/electronic/b278/`.

Biographical Information

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`

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`

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`