ADVANCED CONCEPTS IN TIME-FREQUENCY SIGNAL PROCESSING MADE SIMPLE

Moushumi Zaman, Antonia Papandreou-Suppappola and Andreas Spanias

Abstract — *Time-frequency representations (TFRs) such as the spectrogram are important two-dimensional tools for processing time-varying signals. In this paper, we present the Java software module we developed for the spectrogram implementation together with the associated programming environment. Our aim is to introduce to students the advanced concepts of TFRs at an early stage in their education without requiring a rigorous theoretical background. We developed two sets of exercises using the spectrogram based on signal analysis and speech processing together with on-line evaluation forms to assess student learning experiences. In the paper, we also provide the positive statistical and qualitative feedback we obtained when the Java software and corresponding exercises were used in a signal processing course.*

Index Terms ¾ Java, On-line Exercise, Spectrogram, Time-Frequency Representations.

INTRODUCTION

Classical signal processing tools such as the Fourier transform (FT) are not adequate for analyzing time-varying signals, which are signals whose frequency content changes with time. These signals are important as they naturally occur in many real-world applications [1]; examples include speech, music, biological signals such as dolphin echolocation sounds, biomedical signals such as electrocardiogram (ECG) waveforms, impulse responses of wireless channels, radar and sonar acoustic waves, and seismic acoustic waves.

Time-frequency representations (TFRs) are specifically designed to process time-varying signals as they jointly display time and frequency information demonstrating which frequencies occur at a certain time, or, at which times a certain frequency occurs. They combine time-domain and frequency-domain information by displaying signals over a joint time-frequency (TF) plane [1], [2], [3]. Even though TFRs present the most suitable way to analyze time-varying signals, their complex two-dimensional structure is difficult to implement. In order to understand the theory of these transformations, one requires advanced mathematics and a strong background in signal processing. However, it is necessary to understand these concepts to ascertain which particular TFR is appropriate for a certain application based on the characteristics of the analysis signal [1].

The complex mathematics and theory of TFRs makes it a difficult topic to explain to undergraduate students who do not have the appropriate background. Yet, it is necessary to learn how to process time-varying signals effectively due to their every day application. For example, a musical score is a type of TFR as it represents the change, with time, of the musical notes or frequencies in a song. Similarly, the students could interpret a TFR of a speech segment in order to identify some harmonics in a voice recognition application. With new directions in technological advances, it is becoming increasingly important to understand from an earlier educational stage the importance of theoretical methodology and its effect on widespread applications. This can be achieved using effective teaching techniques such as active learning methodologies with on-line demonstrations. In this paper, we apply such methodologies using modern technology such as the Internet with Java exhibition software to design demonstrations on advanced signal processing TFR applications.

The Java software language [4] was chosen to develop the TFR processing tool for the Internet. Most of the other electronic multimedia-based tools are platform-dependent or delivered using CD ROMs, and are thus not suitable for interactive web-distributed dissemination via the Internet. The attraction of the web for educational purposes is that the web is widespread, flexible, fast, convenient, and interactive. The Internet, together with Java modules, can be used to form object-oriented simulation blocks. The Java software can be used to support a user-friendly and object-oriented simulation environment. Java can run on different operating systems and platforms, which makes it appealing for interactive web applications. Java is also self-contained; there is no need for any external data or program source for an application to run. Any browser with a Java Virtual Machine is capable to run Java applications. Specifically, the Java Virtual Machine interprets Java programs and converts them so that they can be run from that computer [5]. As a result, Java is also faster than programs running from remote servers. Java applets also offer better user interaction. They can enable unified simulations over the Internet, have capabilities for seamless integration of theory and on-line visualization modules, and exploit modern browser technologies to facilitate customized student learning with immediate feedback.

The authors are with the Department of Electrical Engineering, Arizona State University, Tempe, AZ 85287.
(Emails: moushumi@asu.edu, papandreou@asu.edu, spanias@asu.edu)
This work was supported by the National Science Foundation grant CCLI DUE – 0089075.

Among the many TFRs developed in the literature, the spectrogram, a quadratic TFR, is a popular signal processing tool, especially for its application in speech processing. In this paper, we present the Java software modules we developed for the spectrogram together with the associated programming environment, exercises and initial assessment.

THE SPECTROGRAM TFR

The concept of processing in the TF domain dates as early as 1932 with the introduction of the Wigner distribution quadratic TFR. In 1946, the sound spectrograph, that relates to the spectrogram TFR, was introduced to represent speech signals for visual interpretation [6]. In 1966, Cohen's class of TFRs was provided based on the Wigner distribution and its importance in signal processing [2]. A Cohen's class TFR, *Tx(t, f)*, of a signal *x(t)*, is defined as

$$Tx(t,f)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x(\hat{t}+\frac{t}{2})\, x^{*}(\hat{t}-\frac{t}{2})\, e^{-j2p\, ft}\, \Psi_T(t-\hat{t},t)\; d\hat{t}\, dt \qquad (1)$$

where $\Psi_T(t,t)$ is the kernel representing the TFR. Among all Cohen's class TFRs, the Wigner distribution and the spectrogram are most often used for TF analysis due to their simplicity in theory and implementation.

The spectrogram is a Cohen's class TFR in (1) with $\Psi_S(t,t)=h(-t-\frac{t}{2})\, h^{*}(-t+\frac{t}{2})$. It is defined as [7] [8]

$$S_x(t,f)=\left|\int_{-\infty}^{\infty} x(t)h^{*}(t-t)e^{-j2pft}\,dt\right|^2 \qquad (2)$$

where *x(t)* is the signal and *h(t)* is the analysis window. The spectrogram represents the time-varying spectrum of a signal as it computes the frequency content of the signal in short intervals. Specifically, it provides information on which frequencies are present in a signal and how those frequencies vary with time. The spectrogram is simply the squared magnitude of the short time FT, which divides the signal into successive short time segments by the sliding window *h(t)* and determines the local spectrum by means of a FT of the segment. It is based on the assumption that the signal is stationary over the short duration of the segment.

The duration of the window affects the TF localization offered by the spectrogram in time and frequency. Furthermore, a trade-off exists between time and frequency localization because no window exists that is perfectly concentrated in both the time and frequency domains due to the Uncertainty Principle [9]. Specifically, a longer analysis window yields poorer time resolution but better frequency localization; conversely, a shorter analysis window yields better time localization but poorer frequency localization. In the case of a sinusoid signal, for example, a longer window is preferable as it offers better frequency localization to detect the single sinusoidal frequency.

The spectrogram is widely used in many applications such as the analysis and classification of time-varying signals, speech recognition, linguistics, detection of volcanic tremors, and measurement of the filter response of high frequency receivers.

JAVA IN EDUCATION

Java modules have been used extensively in education, especially for math, science, technology and engineering. As Java applets are interactive, students can effectively use them to learn an otherwise difficult topic by visualizing its intermediate steps [10]. Applications range from basic geometry and calculus to advanced mathematical concepts, physics laws, basic quantum mechanics, and emission of radiation [11]. Engineering applets have been developed in many areas including particle filters with robots [12], adaptive filtering [13], antenna arrays [14], discrete cosine transform (DCT) image processing, and filters by windowing [15].

The Arizona State University (ASU) Multidisciplinary Initiative on Distance Learning (MIDL) laboratory developed a Java-based object-oriented programming environment which is called J-DSP [16]. This is available on-line (http://jdsp.asu.edu/) and can be used via the Internet without any installation. J-DSP allows students to set up and execute digital signal processing and communication simulations from any computer that is Internet ready with a simple browser. The various functions are embedded as graphical blocks that can be connected to perform various simulations. The output is viewed by simply double clicking on the block and viewing the dialog box. Some of the J-DSP functionalities include: filter design, fast FT (FFT), and autocorrelation computations.

JAVA IMPLEMENTATION OF THE SPECTROGRAM

We implemented the spectrogram in (2) using Java software in order to make it easier to explain and use by students. Specifically, we designed and integrated it as part of J-DSP and made use of existing functionalities such as signal generators. A snapshot of the spectrogram in J-DSP is shown in Figure 1 where the signal generator block is connected to the spectrogram and plot blocks. The dialog boxes of the plot and spectrogram blocks are also shown in Figure 1.

We implemented the spectrogram with the following features. Different signals, such as sinusoids, linear chirps, and Gaussians, corresponding to *x(t)* in (2), can be chosen from the signal generator block to compute their spectrogram. The user may enter the window length and window type of the spectrogram. The window types available, *h(t)* in (2), are: Hamming, Hanning, Rectangular, Gaussian, Blackman, Bartlett (triangular), and Kaiser. The user can also enter the number of FFT points needed for the FT in (2), and the amount of desirable resolution. Here, resolution 1 implies that all computed samples will be plotted whereas resolution 2, implies that only alternate samples will be plotted. For the image plot, the time axis (with time samples or seconds as units) is horizontal and the frequency axis (with normalized frequency or Hz as units) is vertical; the plot offers a three-dimensional view of the analysis spectrogram, where the shades of red mean increased energy along the frequency axis, and the shades of blue mean decreased energy. The user can choose whether to view the plot in a linear or logarithmic (in dB) scale.

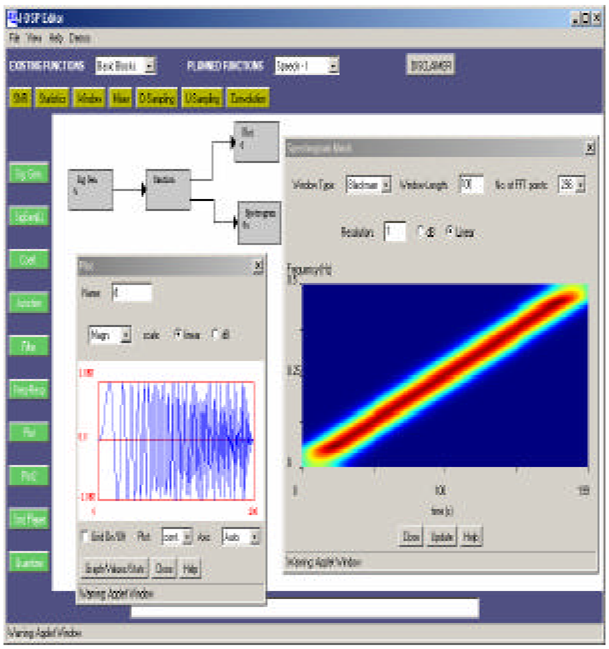

FIGURE 1
A SNAPSHOT OF THE SPECTROGRAM IN J-DSP.

The graphical user interface of the spectrogram block is shown in Figure 2. For this example, the dialog box shows that the number of FFT points is 256, the window type is Hamming and 128 samples long, the view mode is linear, and the resolution is 2. The buttons below the plot are to close the dialog box window, update the spectrogram parameters, and open the help screen.

For detailed observation of the output, we have implemented zooming capabilities to the spectrogram image plot. By clicking on the image plot, a specific area of the plot can be selected to zoom in. To go back to the original plot, the user simply clicks on the zoomed plot. Also, placing the cursor on the plot, the normalized magnitude, and the coordinates of any particular point, can be viewed.

The spectrogram presented some computational and mostly graphical complexities. The three-dimensional image plotting technique is unique as opposed to traditional one-dimensional and two-dimensional plots. Instead of just adding points to create a line, each point is actually drawn as a small rectangle, filled with the color that corresponds to the segment energy. This creates a timing challenge as it takes a significantly longer time to draw the color filled rectangles as compared to drawing single pixels. We employed several programming and optimization techniques to make the function time efficient. This allows the spectrogram of progressive speech to be viewed frame by frame without any delay. With the resolution parameter, a smaller number of points can be plotted to further reduce plotting time, which allows fast operation on slower computers. Note however, that this can decrease the sharpness and readability of the plot.

The spectrogram block can be connected to any one of the two signal generator blocks available. One of the signal generator blocks offers different deterministic and random signals, while the other offers real-life signals such as speech and music. We developed and added the linear chirp signal and the Gaussian signal to the signal generator block. Also, we included the capability to add two or more signals simultaneously and observe the spectrogram of the composite signal. We also made it possible to time shift the signal. Note that this was important in order to demonstrate that the spectrogram preserves time shifts on the analysis signal.

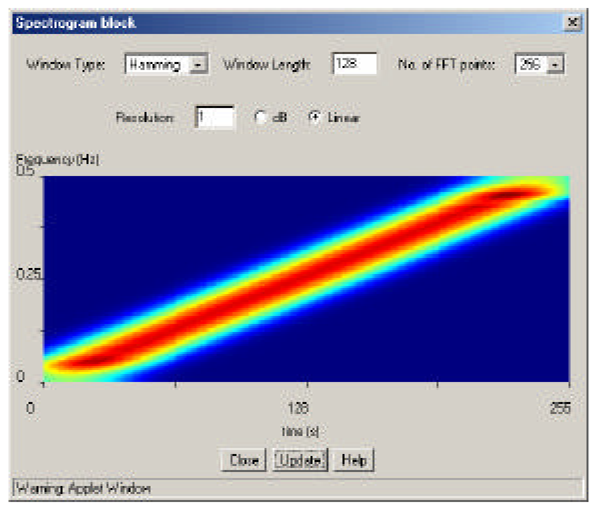

FIGURE 2
GRAPHICAL USER INTERFACE OF THE SPECTROGRAM BLOCK.

The spectrogram in Figure 2 is of a linear chirp signal. As expected, we can see that the signal frequencies increase linearly with time. An example of the spectrogram of two sinusoid signals is shown in Figure 3. Here, the lower frequency sinusoid is shifted in time. Figure 4 shows the spectrogram of two linear chirp signals, which cross in the TF plane as their chirp rates have opposite signs.

The speech and music files are generated progressively one frame at a time. The spectrogram of these types of signals is also generated the same way, and the frames can be viewed one at a time. Figures 5 and 6 show the spectrogram of segments of male and female speech files, respectively. It can be observed that the female speech has fewer frequency bands, implying that the male voice has more pure tones than the female voice.

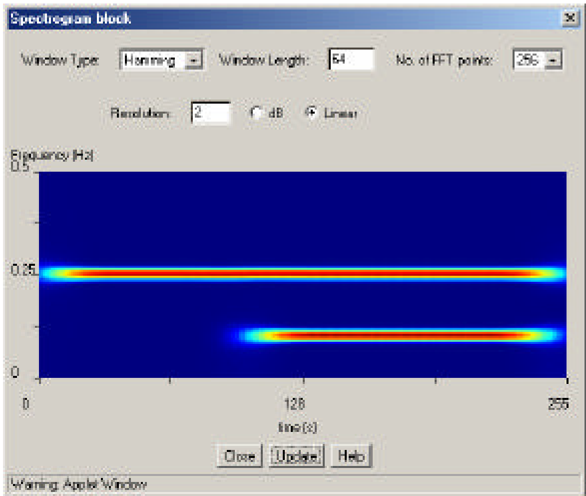

FIGURE 3
SPECTROGRAM OF TWO SINUSOID SIGNALS.

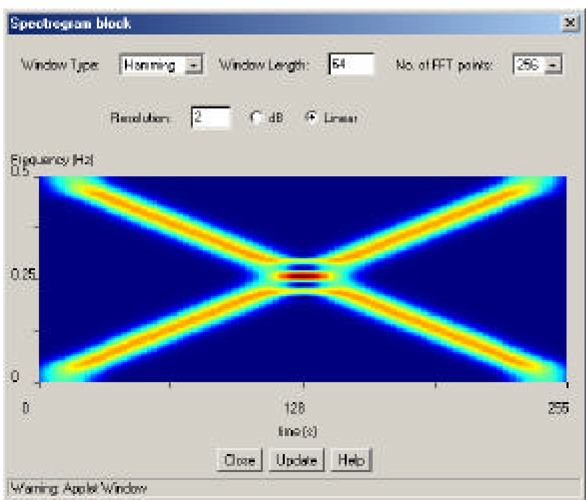

FIGURE 4
SPECTROGRAM OF TWO CROSSING LINEAR CHIRP SIGNALS.

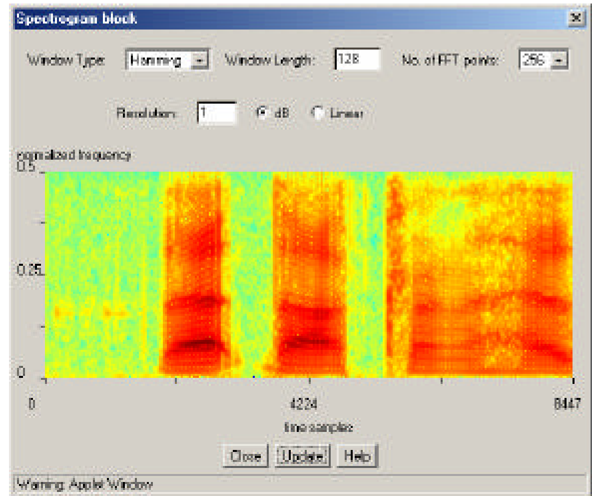

FIGURE 5
SPECTROGRAM OF A MALE SPEECH FILE.

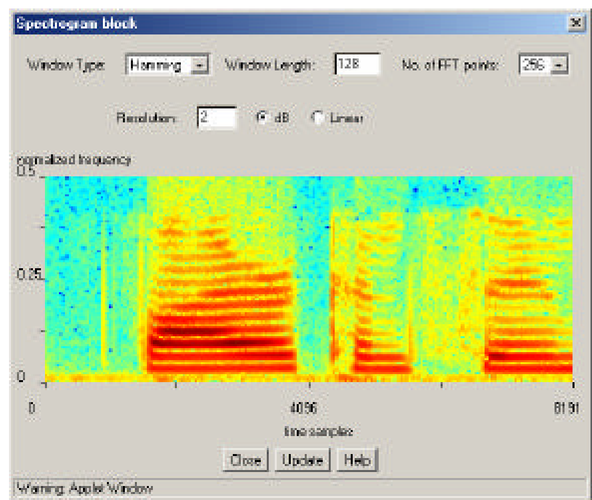

FIGURE 6
SPECTROGRAM OF A FEMALE SPEECH FILE.

SPECTROGRAM EXERCISES

We developed two sets of exercises using the spectrogram. The first exercise concentrates on TF signal analysis, TF localization and choice of spectrogram window, spectrogram and signals in noise, and maximum likelihood estimation (MLE) of unknown sinusoidal parameters with the spectrogram. The second exercise is on speech processing with simulated and real speech, providing a basic understanding of the visual representation of speech.

The first spectrogram exercise consists of four parts:

- **Time frequency localization and choice of window**: In this exercise, we study the TF localization of the spectrogram with respect to the choice of the analysis window.
- **Spectrogram as an analysis tool:** Many quadratic TFRs like the Wigner distribution suffer from cross terms when analyzing multi-component signals or inner interference terms when analyzing mono-component signals with nonlinear TF structures. This exercise demonstrates that the spectrogram is a smoothed version of the Wigner distribution designed to attenuate cross or interference terms. If two signals do not overlap in the TF plane, then the spectrogram smoothes out cross or interference terms using windowing techniques.
- **Spectrogram and signals in noise:** In this exercise, we investigate the use of the spectrogram to detect various signals in additive white Gaussian noise (AWGN).
- **MLE estimation of unknown sinusoidal parameters with the spectrogram:** The MLEs of the unknown frequency and time of arrival parameters of a sinusoid

in AWGN are directly related to the signal's spectrogram. This exercise shows that the location of the peak of the spectrogram in the TF plane yields the estimated parameters. Figure 7 shows the spectrogram of a sinusoid in AWGN. The time of arrival and the frequency of the sinusoid can be clearly observed and estimated from the plot. This is useful for many practical communication, sonar, and radar applications where, for example, knowledge of the sinusoidal frequency provides information on the transmitted signal.

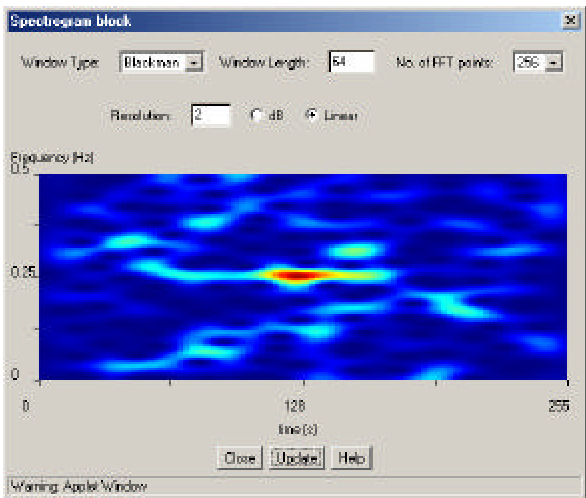

FIGURE 7
MLE ESTIMATION OF SINUSOIDAL SIGNAL PARAMETERS IN AWGN.

As human perception is based on frequency analysis, speech processing is preferably performed in the frequency domain. Unlike stationary signals, speech has a frequency spectrum that varies with time and thus it can be ideally processed in the TF domain using, for example, the spectrogram TFR. The speech processing exercise has two parts:

- **Simulated speech:** This exercise performs fequency estimation of simulated speech using sinusoid signals from the signal generator block. Spectrally shaped white noise is simulated and its spectrogram is observed. The exercise aims to estimate the frequencies and identify the type of speech simulated.
- **Speech files**: Various speech files are analyzed in this exercise as a basic introduction to speech processing. Some of the various topics dealt with in this exercise are the speech and no-speech regions, voiced and unvoiced regions, difference in energy distribution for voiced and unvoiced sounds, number of voiced segments in a specific speech file, highest energy amplitude of the segment, time length and number of harmonics in a voiced segment, formants, formant transitions, narrowband and wideband spectrograms and their relative advantages, and comparison of male and female speech.

EVALUATION FEEDBACK FROM USERS

The user evaluation for the spectrogram exercises is obtained by means of on-line forms. We have developed the electronic forms for the evaluation of the software and the on-line exercises. Qualitative as well as quantitative data is collected automatically and stored on the network. General assessment includes providing feedback on the usability and usefulness of the functions we designed, while specific forms focus on each exercise by posing questions to determine whether the student has learned a concept. Some examples of the evaluation questions are listed in Figure 8 and Table I. The responses were obtained from students taking the Time-Varying Signal Processing course at ASU in the Fall 2002 semester. There were 23 participants.

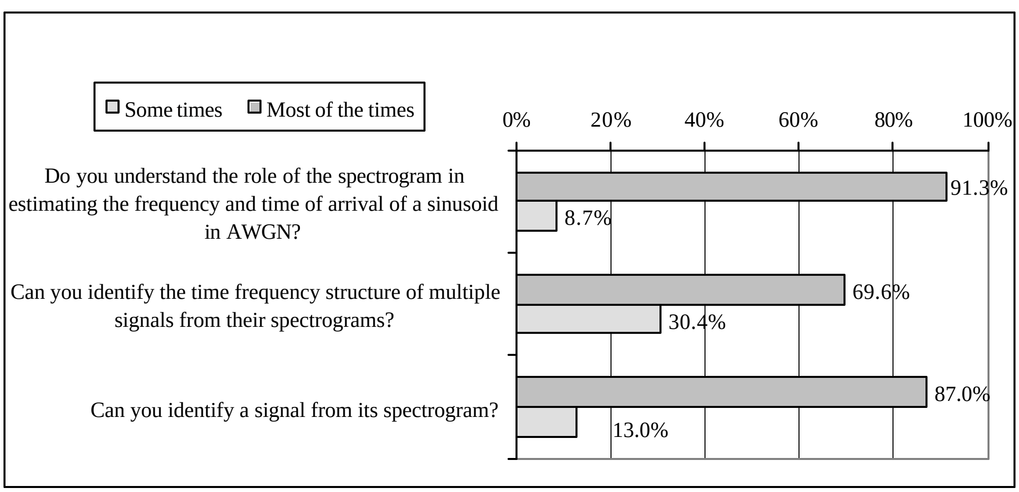

FIGURE 8
USER EVALUATION TO ASSESS THE UNDERSTANDING OF THE CONCEPTS AFTER PERFORMING THE EXERCISES.

TABLE I
USER EVALUATION TO ASSESS THE UNDERSTANDING OF THE CONCEPTS AFTER PERFORMING THE EXERCISES.

Evaluation questions	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
1. Performing the exercise, you can now understand whether a shorter or longer spectrogram window yields better time localizat ion and why.	56.5%	43.5%	0%	0%	0%
2. The exercise improved your understanding of the time frequency structure of chirp signals.	52.2%	39.1%	0%	8.7%	0%
3. Setting up the required simulations was pretty easy	60.9%	30.1%	9%	0%	0%

Users provided valuable feedback by answering a comprehensive set of questions. All of the students agreed that they were more comfortable with the topics after completing the exercises. From Figure 8, more than 90% of the students understood the role of the spectrogram in estimating the frequency and time of arrival of a sinusoid in noise; 69.6% reported that it was now easier to identify the TF structure of multiple signals from their spectrograms. From Table I, all of the students gained understanding of spectrogram localization trade-offs; 91.3% of the students felt that the exercises improved their understanding of the TF structure of chirp signals. 60% of the students did not even need the help screens to perform the simulations, while 30% of the students found the help screens to be sufficient and helpful (not shown in table). Note that space was also provided to the students to type comments and suggestions regarding the exercises. Some students, for example, suggested adding more TFRs such as the Wigner distribution. The exercises and evaluation forms can be accessed through the J-DSP web site http://jdsp.asu.edu/.

CONCLUSION

In this paper, we proposed to use the Internet with Java exhibition software for teaching advanced signal processing methodologies without rigorous mathematics. Specifically, we designed a spectrogram TFR module to analyze time-varying signals such as speech. Our evaluations have shown that this is a promising and effective means of introducing advanced TF signal processing techniques to students. Future plans include lab exercises for the Digital Signal Processing course and Speech Processing course. Currently, more simplified versions of the exercises are being formulated as demonstrations for use in the WISE-Up Women in Applied Science and Engineering summer program at ASU that aims to attract pre-college female students to the engineering discipline.

REFERENCES

[1] Papandreou-Suppappola, A. ed., "Applications in Time-Frequency Signal Processing", Boca Raton, Florida: CRC Press, 2002.

[2] Cohen, L., "Time-Frequency Analysis", Englewood Cliffs, NJ: Prentice Hall, 1995.

[3] Flandrin, P., "Time-frequency/Time-scale Analysis", New York: Academic Press, 1999.

[4] Schildt, H., Naughton, P., "Java™ 2: The Complete Reference", Third Edition, New York: McGraw-Hill, 2000.

[5] "How the Internet works", http://www.smartcomputing.com/.

[6] Koenig, W., et al, "The sound spectrograph", *Journal of the Acoustical Society of America*, vol. 18, 1946, pp. 19-49.

[7] Allen, J.B. and Rabiner, L.R, "A unified approach to STFT analysis and synthesis", *Proc. of the IEEE*, vol. 65, Nov. 1977, pp. 1558-1564.

[8] Altes, R. A., "Detection, estimation, and classification with spectrograms", *Journal of the Acoustical Society of America*, vol. 67, April 1980, pp. 1232-1246.

[9] Hlawatsch, F., Boudreaux-Bartels, G.F., "Linear and quadratic time-frequency signal representations", *IEEE Signal Processing Magazine*, vol. 9, April 1992, pp. 21-67.

[10] Kamthan, P., "Java applets in Education", http://tech.irt.org/articles/js151/

[11] The EOE Foundation web page, http://www.eoe.org/

[12] "Particle filter demo", http://www.cs.washington.edu/ai/Mobile_Robotics/mcl/

[13] Harteneck, M., Adaptive Filtering, http://www.spd.eee.strath.ac.uk/~moritz/

[14] Barrus, K., Antenna Array, http://www.nwlink.com/~klbarrus/antenna/antenna.htm

[15] Wu, O., Filters by windowing, http://web.mit.edu/6.555/www/fir.html

[16] Spanias, et al, "On-line laboratories for speech and image processing and for communication systems using J-DSP", *10th IEEE DSP Workshop and the 2nd IEEE Workshop on Signal Processing Education*, Atlanta, October 2002.