Hands-on exposure to signal processing concepts using the SPC toolbox

Fargues, Monique P.

http://hdl.handle.net/10945/41124
Abstract—We present an interactive software package developed to complement and facilitate the teaching of digital signal processing (DSP) concepts in various courses. The Signal Processing and Communications (SPC) software takes advantage of graphical user interface (GUI) capabilities contained in MATLAB®. SPC is a window-based, user-friendly tool that allows users to develop digital filters, analyze signals, and easily design basic signal modeling tools without requiring the user to have strong coding abilities or be very familiar with MATLAB. To illustrate how SPC can be used in the classroom, we present applications of the SPC Toolbox that were used in two courses at the Naval Postgraduate School—Discrete System and Digital Signal Processing.

I. INTRODUCTION

DIGITAL Signal Processing (DSP) is a continuously changing field whose emphasis has increased tremendously in academic institutions in the last 10 years. While DSP is quite important in electrical engineering education, the mathematical concepts used can be rather dry and difficult when presented in the classical classroom environment. These concepts become more meaningful to the student when they are actually applied to specific problems. Thus, numerous innovative proposals have been presented in the last few years to emphasize DSP applications in the classroom and to provide a better "hands-on" approach to the teaching of DSP concepts [1], [2].

The Signal Processing and Communications (SPC) software project first started in 1993 in the Electrical and Computer Engineering Department of the Naval Postgraduate School (NPS). It was intended to provide a basic set of DSP tools that facilitate the application of various DSP techniques [3], [4]. The typical student pursuing a Masters degree at NPS is different from the typical graduate student studying at a civilian university: the NPS student is older; has five to nine years of work experience; and has been away from the university environment for an equal amount of time. In addition, his or her undergraduate major may be in a discipline other than electrical engineering. Thus, NPS students studying for a Master's Degree in electrical and computer engineering are usually required to take several undergraduate courses to prepare themselves for graduate courses. As a result of their very diverse backgrounds, NPS students favor hands-on approaches to problems. This allows them to better visualize the meanings of theoretical concepts and how they can be applied in practice. Therefore, courses taught in the Digital Signal Processing track in the Electrical and Computer Engineering Department at NPS rely heavily on projects and computer-based assignments in an effort to allow the student to make a better connection between theory and application. Furthermore, NPS students at the beginning of their curriculum have had little or no experience with scientific programming. Thus, SPC was designed to allow students without strong coding abilities to visualize concepts through various applications, and for this purpose graphical and sound capabilities represent an integral part of the software.

The main motivation behind the development of the SPC Toolbox was to design a computer environment providing a basic set of DSP tools allowing students to:

- Easily examine signals in time and frequency using a wide range of techniques.
- Easily perform fundamental signal processing operations (such as filtering) on signals.
- Easily process signals using more advanced techniques.
- Study fundamental concepts of DSP.
- Minimize the amount of coding involved.

It was of the utmost importance that the workstation's software be user-friendly, affordable, and function on multiple platforms, i.e., be easily transported between different types of computers. (At NPS, SPC was successfully tested and installed on IBM PC's and SUN workstations.) In addition, SPC relies heavily on graphical user interfaces (GUI's). Most software considered user-friendly today is based on GUI interfaces offering pull-down and pop-up menus, push buttons, check boxes, and other graphical controls that can be activated with a pointing device such as a mouse to select various options and execute commands [5]. The use of a GUI interface relieves the user from having to memorize cryptic textual commands—often with a large number of even more cryptic options available. In addition, a well-designed GUI interface usually has the added benefit of reducing the amount of time a user spends learning how to use an application. Finally, if a consistent GUI interface is designed across applications, it allows the user to more quickly use another application with a similar GUI interface.

SPC is a MATLAB-based set of individual tools that are linked together so that different operations can be cascaded.
MATLAB version 4.2 and its Signal Processing Toolbox, available from The MathWorks, are required to run the full capabilities of the SPC software. SPC functions are divided into five main groups: time-domain signal analysis; filtering; spectral estimation and signal modeling; communication signal generation; and graphical tools. SPC allows the user to develop digital filters and to visualize their effects, to analyze speech signals, and to design basic parametric autoregressive (AR), moving average (MA) and autoregressive, and moving average (ARMA) techniques. In addition, numerous graphical tools are provided that allow the user to zoom in on a specified section of a given signal, do cut-and-copy operations, etc. Section II contains an overview of the capabilities of the software. SPC has been used in several DSP courses taught in the Electrical and Computer Engineering Department at NPS, from basic DSP courses to more advanced statistical signal processing and speech signal processing courses. Section III describes the use of SPC in basic filter design, as taught in our undergraduate course Introduction to Discrete Systems. Section IV describes the use of SPC in the first digital signal processing course. Finally, conclusions are presented in Section V.

II. SPC SOFTWARE OVERVIEW

A. Software Description

The SPC Toolbox provides a number of commands that implement a variety of digital signal processing and communication techniques. All of the routines use MATLAB (version 4.0–4.2) m-files with some functions implemented as both m-files and mex-files (written in the C-programming language). In addition, we developed a new MATLAB paradigm called “RunTime” to integrate the separate SPC GUI tools and to provide the ability to run multiple copies of the same tools [3], [4], [6]. Most of the SPC commands are stand-alone or rely on other SPC Toolbox routines. However, some commands, such as those implementing FIR windows, FIR filter design, and digital filter design of analog IIR filter prototypes rely on commands from the MATLAB Signal Processing Toolbox.
In addition, a number of graphics-based GUI tools provide an interactive, graphical environment that offers relief from the MATLAB command line prompt. Some of these tools essentially act as front-ends to other functions. Others provide graphical environments to perform tasks not readily available or easily performed using MATLAB commands (i.e., visual editing of signals), thereby relieving users from the requirement to write significant amount of code. The following describes the most significant graphic-based tools provided by SPC. A complete list of SPC tools may be found in the *SPC User’s Guide* [4].

1) **Signal Edit Tool (SIGEDIT):** provides a graphical interface that allows for visual cut-and-paste, copy, amplify, and play (using the MATLAB soundtool command). All commands are mouse-driven and apply directly to the plot of the data set.

2) **Speech Signal Edit Tool (VOICEDIT):** provides a graphical interface that allows the user to compute the zero-crossing rate (ZCR) and the short time energy (STE) of a given data set. The user can select window length, type, and overlap. In addition, various smoothing and median filter options are available to smooth out the resulting ZCR and STE.

3) **Signal Filtering Tool (SIGFILT):** provides a graphical interface that may be used to filter a given data set. This environment allows the user to select the filter type, order, and cutoff frequency regions by dragging vertical cursors that appear in the frequency plot of the given data. Once applied, the filtered signal power information is displayed on top of the filter magnitude response. This capability allows the students to better visualize the effects of filtering in the frequency domain.

4) **Signal Modeling Tool (SIGMODEL):** provides a graphical interface that allows the user to compute AR, MA, and ARMA models for a given data set. The available AR methods are the correlation, the covariance, the modified covariance, and the Burg method. Available MA methods are Prony’s, Shank’s, and Durbin’s method. The user obtains ARMA modeling by selecting both AR and MA options. The tool can be used for synthesis operations, as the user has the option to select a white noise or periodic impulse train to send through the estimated model.

5) **2-D Spectral Estimation Tool (SPECT2D):** provides a graphical interface that allows the user to compute classical spectral estimation techniques such as the pe-
FARGUES AND BROWN: HANDS-ON EXPOSURE TO SIGNAL PROCESSING CONCEPTS USING THE SPC TOOLBOX

Fig. 3. SIGFILT tool window.

riogram, the Daniell, and the Welsh estimator. In addition, two subspace methods are available; minimum variance and MUSIC estimators.

6) **3-D Spectral Estimation Tool (SPECT3D):** provides a graphical interface that allows the user to generate a spectrogram very easily. The user has full control of the time window length, type, and overlap. In addition, SPECT3D uses MATLAB graphical capabilities to create mesh plots using various shading options and viewing orientations. Finally, the user has the option to magnify specific regions of the time/frequency plane to allow for more precise analysis of the data.

7) **Graphical Filter Design Tool (GFILTERD):** provides a graphical interface that allows the user to visualize easily the effects of poles and zeros on a desired filter transfer function. A window displaying three plots is opened when the tool is called. The first graph is a polar plot of the complex plane. The second and the third plots show the magnitude and phase of the filter frequency response obtained after placing poles and zeros in the complex plane. Pole/zero placement, addition, and deletion can be entered on the complex plane graphically using the mouse.

8) **ZOOMTOOL:** provides graphical tools that allow the user to zoom in on certain sections of a plot. In addition, mouse-driven horizontal and vertical cursors are available to the users to identify specific values on the plot. This tool is used by most of the other tools to provide zooming capabilities.

B. Software Access

The SPC Toolbox is available for use on two different platforms: MS-Windows 3.X and UNIX. The software package is made available free, and the latest version of the software and its user’s guide may be obtained at the World-Wide-Web (WWW) site with URL: http://vislab-www.nps.navy.mil/~fargues/spc.html, or by anonymous ftp from ftp.nps.navy.mil in the directory /pub/ece/spctools. We encourage interested readers to obtain the software, and we welcome feedback on its performance.

III. USING SPC IN "DISCRETE SYSTEMS"

In this section, we illustrate how SPC was used in the course titled Discrete Systems (EC2400) during the Spring 1995 quarter. EC2400 is the first linear systems course that is taken by students in the ECE curriculum. This course
presents an introduction to basic tools such as LTI systems, frequency responses and filters, and the Z-transform. For example, students taking this course are expected to gain a clear understanding of the role of poles and zeros in the plot of a filter frequency response. This concept can be easily illustrated using the SPC Toolbox, as shown below.

The specific SPC graphical interface tool used for this course is \textit{GFILTERD}. \textit{GFILTERD} provides an interactive, graphical environment to design digital filters through individual (complex-conjugate pair) placement of poles and zeros in a plot of the complex plane, which represents the complex pole/zero plot of the filter transfer function. Poles and zeros may be placed, moved, and deleted using mouse-driven commands or the keyboard. This tool opens a window containing three plots; the first plot is a polar plot of the complex plane that contains the poles and zeros of the filter. The second and the third plots show the magnitude and the phase of the resulting filter frequency response, as illustrated in Fig. 1. The objectives of the three parts of the project described below are to familiarize the students with the concepts of poles and zeros and how they relate to frequency responses.

\textit{Part A:} Students are asked to study the effects of the locations of single zeros, to move the locations of the zeros, and to investigate the effects. Next, students are asked to do the same for a frequency response with poles only.

\textit{Part B:} Students are asked to increase the order of the zeros and to investigate how the frequency response changes when increasing the order of the zeros. Students are asked to repeat the procedure with a pole-only filter frequency response.

\textit{Part C:} Students are asked to study the effects due to both to poles and zeros on a frequency response and to investigate how the effects change when poles and zeros are moved closer to each other.

\textbf{IV. USING SPC IN "DIGITAL SIGNAL PROCESSING"}

SPC can also be used in the Digital Signal Processing course (EC3400) to graphically design digital filters and to visualize the effects of these filters on the data. The specific SPC graphical interface tools used in the context of this course are \textit{SIGEDIT}, \textit{SIGFILT}, and \textit{SPECT3D}. \textit{SIGEDIT} allows the user to manipulate data files graphically very easily via the use of vertical cursors and pull-down menus, as illustrated in Fig. 2. Fig. 2 also illustrates SPC's capability to launch several graphical interfaces in cascade using the pull-down menu \textit{Launch}, shown open in this figure. \textit{SIGEDIT} can be used to isolate and play, cut, crop or amplify various portions of speech data very easily.

The graphical interface \textit{SIGFILT} relies on the MATLAB Signal Processing Toolbox to compute filtering expressions. However, SPC adds a user-friendly interactive and graphical interface to the process that does not require the students to have familiarity with the MATLAB Signal Processing Toolbox. \textit{SPECT3D} is a graphical interface that allows students to easily visualize spectrograms of various types. The objectives of the project presented below are to familiarize students with the effects due to filtering. Students are asked to digitize a speech sentence that was jammed by one stationary narrowband interference, to remove the interference, and to
transcribe the contents of the sentence. *SIGFILT* opens a window presenting the frequency information of the signal to be filtered and the frequency response for a default filter. Filter types, cutoff frequencies, and other filter characteristics may be adjusted by selecting the desired options available in the various check boxes and pull-down menus. This filtering process is illustrated in Fig. 3, which shows a spectral estimate of the noisy speech signal with its stationary jammer around 1600 Hz. Fig. 3 also shows the frequency response of the Chebyshev stopband filter of order eight selected to remove the jammer. Note that cutoff frequencies may be selected using the keyboard or by moving the vertical cursors in the window. Once the user is satisfied with the filter characteristics, the filter can be applied by clicking on the *Apply* box. After the filter is applied, the *SIGFILT* window contains three curves: the frequency response of the selected filter; the power spectral density of the signal before filtering was applied; and after the signal before filtering was applied. In Fig. 3, noisy and filtered signals are represented with dotted and solid lines, respectively, to illustrate the filtering effects in the frequency domain. The filtered speech signal can then be saved back into the MATLAB workspace using the SPC *save* command available in the *Workspace* pull-down menu. Next, the spectrogram obtained using the SPC tool SPECT3D on a segment of the filtered speech signal is shown in Fig. 4. SPC places slider controls located on the sides of the plot to control the orientation and brightness of the three-dimensional plot. Alternative colormaps and shading may be selected by opening the pull-down menus *Colormap* and *Shading* located at the top of the window.

V. CONCLUSION

This work has described a user-friendly, window-based, interactive software package designed to assist students in the application of various DSP concepts learned in the classroom. Its graphical capabilities facilitate the analysis of data and decrease the programming required for students. We illustrated a few of the softwares capabilities in filter design applications.

REFERENCES


Monique P. Fargues (S’83–M’89–SM’95) was born in Bordeaux, France. She received the Engineer's Degree from the Ecole Superieure d'Ingénieurs en Electronique et Electrotechnique in Paris, France, in 1982. She received the M.S. degree in 1984 and the Ph.D. degree in 1988, both in electrical engineering, from the Virginia Polytechnic Institute and State University, Blacksburg, VA.

She joined the Electrical and Computer Engineering Department at the Naval Postgraduate School in 1989, where she is currently an Associate Professor. Her main research interests include digital and stochastic signal processing, signal modeling, and numerical mathematics.

Dr. Fargues is a member of SIAM, SPIE, and Sigma Xi. She currently serves as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING.

Dennis W. Brown (S’84–M’86) was born in Hutchinson, KS, in 1960. He received the B.S. degree in electrical engineering from Kansas State University in 1986 and the M.S. degree in electrical engineering from the US Naval Postgraduate School in 1995.

He enlisted in the US Marine Corps in 1978 and served as an electronic intelligence analyst until 1982. He was commissioned an officer in the US Navy on July 31, 1987. He is currently serving as the Signal Analysis Laboratory Technical Director for Commander, Naval Security Group Command (Technology Directorate), Fort Meade, MD. His research interests include development of intuitive operator interfaces for signal processing/analysis systems.