AC 2009-2269: DESIGNING EFFECTIVE USER INTERFACES FOR SOFTWARE SIMULATIONS TO TEACH SIGNAL PROCESSING CONCEPTS

Sam Shearman, National Instruments

Sam Shearman is a Senior Product Manager for Signal Processing and Communications at National Instruments (Austin, TX). Working for the firm since 2000, he has served in product management and R&D roles related to signal processing, communications, and measurement. Prior to working with NI, he worked as a technical trade press editor and as a research engineer. As a trade press editor for "Personal Engineering & Instrumentation News," he covered PC-based test and analysis markets. His research engineering work involved embedding microstructures in high-volume plastic coatings for non-imaging optics applications. He received a BS (1993) in electrical engineering from the Georgia Institute of Technology (Atlanta, GA).

Designing Effective User Interfaces for Software Simulations to Teach Signal Processing Concepts

Abstract

Educators have embraced software simulations as a tool for teaching signal processing concepts. Simulations allow students to interact with abstract concepts such as convolution, difference equations, filtering, sampling theory and many more. Software simulations strive to achieve learning objectives by presenting an interactive user interface that gives students the ability to interactively explore relationships. Such interaction seeks to improve learning by fostering an intuitive feel for the relationship between parameters and the results.

It is evident that the design of the user interface plays a key role in success or failure. For instance, students can easily be distracted by elements of the software or activities that are not directly related to the subject at hand.

This paper presents user interface design advice that works toward development of software simulations and related student activities for teaching signal processing that successfully achieve learning objectives. Interactive software demonstrations developed using NI LabVIEW provide examples that support the discussion.

Software for Signal Processing Education

Educators have long realized the value of the PC for signal processing education[1-6]. A standard desktop or laptop PC offers a hardware platform for exploration and design that is widely accessible. Engineering software environments such as National Instruments LabVIEW[7] enable educators and students to create, modify, and interact with custom educational software applications that explore linearity, time invariance, signal representation, transforms, digital filters, sampling theory, convolution, and other signal processing concepts. Such software can act as a tool to demonstrate a concept or as the basis for student exercises. It can be distributed in a variety of ways; on DVDs or through the Internet, sometimes through a convenient browser-based interface[8-9].

This paper focuses on signal processing concept software with graphical user interface (GUI) elements for interactivity. Such software can enable interactive examination of concepts through experimentation in which students specify trials by manipulating controls on the user interface such as knobs, dials, digital controls, switches, and buttons. As the input parameters are changed, the software shows the result by updating indicators on the user interface such as graphs, plots, numeric displays, and LEDs.

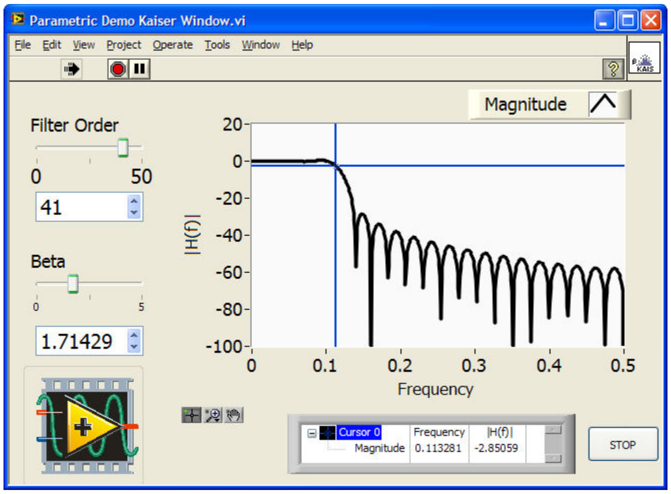

Figure 1, An example signal processing concept demonstration that examines Kaiser-Bessel window FIR filter design.

Figure 1 shows an example signal processing concept demonstration built using LabVIEW. The example explores Kaiser-Bessel window FIR filter design, examining the relationship between filter design parameters and magnitude response. Controls on the left allow the student to modify filter order and beta values to specify a trial low-pass filter design. A graph on the right dynamically plots the magnitude response of the design in decibels as a function of normalized frequency. Exploration is possible on the graph, in that the user can zoom in / out by modifying the ranges of the plot axes. Doing so enables the user to see details in the magnitude response such as the passband ripple. The graph also includes a cursor that can be moved along the magnitude response trace to find, for instance, the -3 dB frequency.

User Interface Design Choices

User interface design involves many choices. You can choose from among a variety of controls and indicators—for instance, LabVIEW includes knobs, dials, sliders, digital inputs, plots, XY graphs, and many others. Building a custom user interface, you are free to choose any number of these, arrange them arbitrarily and assign various colors / fonts.

Given all of these freedoms, it is not surprising that some signal processing demonstration software does a better job than others. In developing this type of software for presentations. training, and other needs, we have found that the design of the user interface plays a significant role in determining effectiveness of a demo in improving understanding. The most successful present a UI that encourages exploration, motivates learning, and facilitates intuitive understanding. Careless design can easily result in something that does a better job of distracting or confusing a student than it does in teaching.

The importance of UI design for improving software and Web sites that involve human-computer interaction is widely recognized in software engineering ideas and methodology related to usability[10-11]. Researchers have considered the design and effectiveness of interactive educational simulations in the context of physics, chemistry and physical science[12-13]. These efforts, our experience, and some common sense provide the basis for specific UI characteristics that lead to success.

The Value of Interactivity with Immediate Feedback

Interactive user interfaces can encourage intuitive understanding by enabling qualitative exploration and providing immediate feedback. As the user modifies input controls, the software rapidly calculates and displays the result by refreshing graphs and numeric displays. Seeing the result appear under their direction engages the user and encourages exploration by providing on-screen action that grabs their attention. When results are presented on plots and graphs, the presentation can give users the feeling that they are controlling an animation.

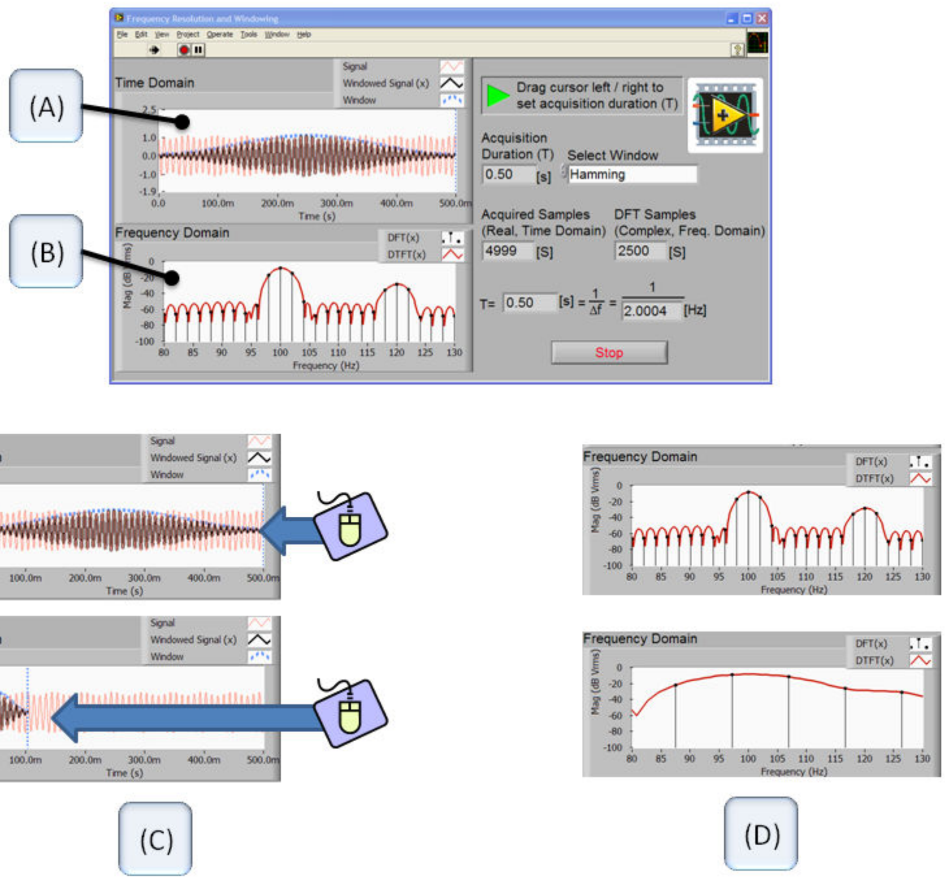

Figure 2, A signal processing concept demonstration that explores the relationship between frame / block length, windowing, and DFT frequency resolution.

As an example, consider a signal processing concept demonstration (Figure 2) intended to give the user an intuitive understanding of the relationship between the frequency resolution of a discrete Fourier transform (DFT), the frame/block length, and choice of window. The demonstration calculates the DFT of a user-specified portion of a test signal that is the sum of two sinusoids with closely spaced frequencies. A graph in the upper left of the UI (Figure 2A) plots the test signal and provides a cursor control that the user can move to specify the frame length. A drop-down selector allows the user to choose among a variety of windows to scale the frame prior to calculating the DFT. As the user moves the cursor or selects a new window, a graph in the lower left (Figure 2C) refreshes to show the resulting DFT. The user can gauge the frequency resolution of the DFT by looking for the peaks that correspond to the two frequency components of the input signal (Figure 2D). With the cursor positioned all the way to the right, the user will see two distinct peaks. As they move the cursor left to decrease the frame length, they will see the peaks become less distinct, eventually blending together.

UI-based approaches that present interactivity with immediate feedback contrast alternatives that involve working with a high-level text-based language such as .m file script. Much of the interactivity of such approaches involves typing in an environment built around a text editor to modify and execute scripts. This type of interactivity can discourage intuitive understanding because the user is often forced to change their focus multiple times as they explore a concept. For instance, after executing a script that examines a concept, the user might next need to modify the script to modify a parameter in order to further explore the concept. Returning to a text editor to make the change forces the student to shift focus away from the concept at hand.

Design for Exploration

An approach to UI design that encourages exploration involves working to tailor the UI to suit a specific sequence of activities. To do so, you can write out a step-by-step script that describes what the user does, how the software responds, what insight is gained, and other details. Such a script can provide an organizational framework that helps to focus your efforts.

Figure 3 shows an interactive concept demonstration that examines sampling theory and aliasing. Its UI reflects a number of design decisions based on a sample script shown in Table 1. As you might expect, the parameters and displays mentioned in the steps have corresponding user interface elements. Input controls on the left allow the user to set the input signal frequency and sampling rate. Given these inputs, the demonstration code synthesizes several signals that are displayed using the graph indicators on the right. The top graph shows a time-domain view that can display the input signal, the signal samples and an alias. The lower graph shows the magnitude spectrum of the input signal and the sampled signal.

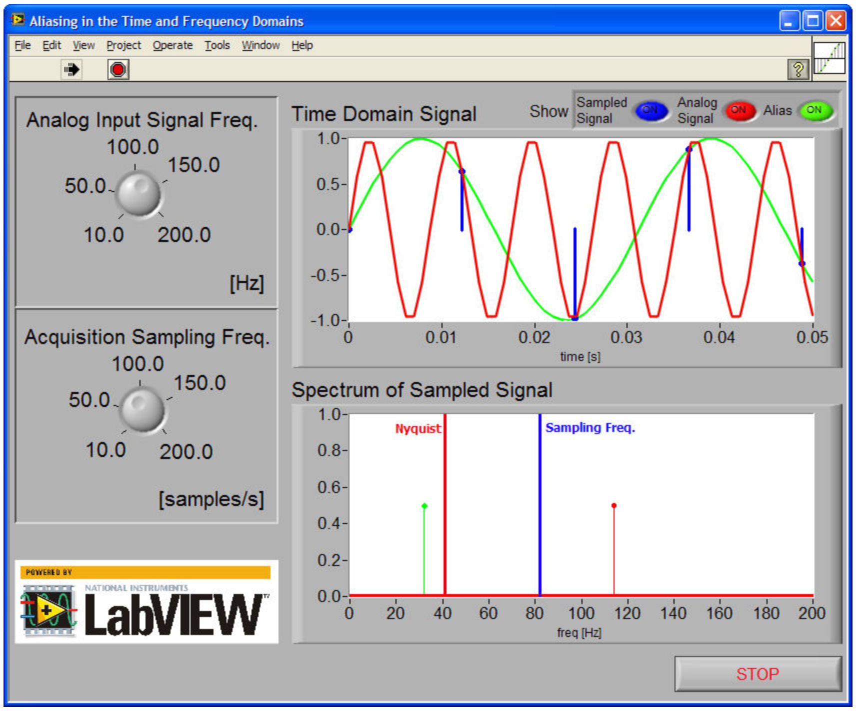

Figure 3, An interactive concept demonstration that explores sampling theory and aliasing.

	Task / Action	Purpose / Intent
1.	The user specifies the frequency of an input sinusoidal signal that will be sampled. Several graphs update to show time- and frequency-domain views of the input signal.	Watching the input signal graphs as they change the frequency, the user understands link between the frequency control and the signal and realizes that they can dynamically interact with the graph.
2.	The user specifies a sampling rate greater than the Nyquist rate for the input signal and enables sampling. Several graphs update to show time- and frequency-domain views of both the input signal and a sampled version of it.	Examining the time- and frequency-domain plots of the input and sampled signal, the user sees that the sampled signal does a good job of representing the input signal.
3.	The user sets the input sinusoid frequency to a value that exceeds the Nyquist rate. Several graphs update to show time- and frequency-domain views of both the input signal and a sampled version of it.	Examining the time-domain plot of the input and sampled signal, the user sees that the samples still correspond to points on the input signal. Examining the frequency-domain plot, they see that the input signal peak now differs from the sampled signal peak.

	Task / Action	Purpose / Intent
4.	The user enables a display of the alias on the time-domain plot.	Seeing the alias plotted together with the input signal and signal samples, the user realizes that the signal samples can be interpreted as an alias.

Table 1, An example script describes the sequence of some of the activities that you might expect as a user works with a concept demo that explores sampling theory and aliasing.

The sequence of activities leads to some guidance involving the position of the controls and indicators. Arrangement of user interface elements can be used to implicitly guide a user through their exploration. Notice that in the example of Figure 3, we have intentionally placed the signal frequency control in the upper left as it is the first control that we would like the student to find and interact with. The upper left corner is a natural starting point that is in line with expectations set by the left-to-right, top-to-bottom reading order for English and other languages.

Control Choice Sets the Action

Other UI design choices involve selection of appropriate controls and indicators types and, as mentioned earlier, there are many alternatives. For setting numerical values LabVIEW includes knobs, dials, sliders, basic digital numeric, and many others. The variety gives you alternatives for the action necessary to change the value. For instance, basic digital controls are useful when you want the user to type an exact value. Knobs, dials and sliders sacrifice such precision, but allow users to modify values more actively and continuously by clicking and dragging with the mouse.

The example shown in Figure 3 uses knobs for the input signal frequency and sampling rate parameters. The intent of this selection is to encourage the user to actively change the value for qualitative exploration. Because the graphs refresh immediately as the inputs change, the user can animate the display by turning a knob. Such immediate feedback can engage the user and helps intuitive understanding.

For example, the user can turn the knob to increase the input signal frequency past the Nyquist rate. Watching the frequency-domain plot, they will see the input signal peak cross the Nyquist rate and continue to increase as an alias peak breaks off and starts to decrease. At the same time, the time domain plot will show the input signal, the alias and the location of the signal samples. Plotting the three signals together emphasizes how an alias can be thought of as an alternative interpretation of the signal samples.

Digital numeric controls could have been chosen as alternatives to knobs for the input signal frequency and sampling rate parameters. Such controls allow users to set specific values through keyboard entry. This type of action can be useful if the exercise that you plan for your demonstration is quantitative and requires entry of precise values—e.g. "set the input signal frequency to 400 Hz." Keyboard entry though does not provide the continuous active engagement that is possible when working with the mouse. To enable both qualitative and quantitative interaction, you can choose a combined control, such as the combined slider / digital controls used for the Filter Order and Beta parameters in Figure 1.

Conclusion

UI design for interactive signal processing concept demonstration software involves choices that affect usability. Putting together a script that relates how a user might interact can provide a framework helps with choices such as type and arrangement of UI elements.

Bibliography

[1] S.H. Mousavinezhad, "Computerized Tools in Digital Signal Processing," 2003 ASEE Annual Conference and Exposition Proceedings, Washington, D.C., June 1996, pp. 1532-1535, http://soa.asee.org/paper/conference/paper-view.cfm?id=12540.

[2] James H. McClellan and Jordan Rosenthal, "Animating Theoretical Concepts for Signal Processing Courses," 2002 ASEE Annual Conference and Exposition Proceedings, Montréal, 2002, http://soa.asee.org/paper/conference/paper-view.cfm?id=17051.

[3] Murat Tanyel, "Enhancing the DSP Toolkit of LabVIEW," 2002 ASEE Annual Conference and Exposition Proceedings, Salt Lake City, 2002, http://soa.asee.org/paper/conference/paper-view.cfm?id=16908.

[4] Murat Tanyel, "Explorations In Communication Systems Using a Virtual Toolkit," 2003 ASEE Annual Conference and Exposition Proceedings, Salt Lake City, 2003, http://soa.asee.org/paper/conference/paper-view.cfm?id=18001.

[5] S. Easwaran, "An Innovative Software Tool for Teaching Discrete Convolution from the Perspective of the Output Signal in Digital Signal Processing," 2005 ASEE Annual Conference and Exposition Proceedings, Portland, 2005, http://soa.asee.org/paper/conference/paper-view.cfm?id=20967.

[6] Murat Tanyel, "Virtual Toolkit As a Tool for Innovation," 2006 ASEE Annual Conference and Exposition Proceedings, Chicago, 2006, http://soa.asee.org/paper/conference/paper-view.cfm?id=1016.

[7] Marlin Viss and Murat Tanyel, "From Block Diagram to Graphical Programs in DSP," 2001 ASEE Annual Conference and Exposition Proceedings, 2001, http://soa.asee.org/paper/conference/paper-view.cfm?id=15905.

[8] James H. McClellan, Ronald W. Schafer, and Mark Yoder. (2003) Signal Processing First-- 2nd Ed., Companion examples / labs, http://www.rose-hulman.edu/dspfirst/visible3/contents/index.htm?page=./contents/demosLV.htm.

[9] Erik Luther and Jim Cahow. (2007, Aug.) Interactive Digital Filter Design - Online Tool for IIR Filter and FIR Filter Design, http://cnx.org/content/m13115.

[10] Jakob Nielsen, Usability Engineering. San Francisco: Morgan Kaufmann, 1993.

[11] Usability.gov, http://www.usability.gov.

[12] W. K. Adams et al., "A Study of Educational Simulations Part I – Engagement and Learning," Journal of Interactive Learning Research, pp. 397-419, July 2008.

[13] W. K. Adams et al., "A Study of Educational Simulations Part II - Interface Design," Journal of Interactive Learning Research, Oct. 2008.