Interactive Simulations in Signals and Systems

Anthony Richardson
University of Evansville, richardson.tony@gmail.com

Interactive Simulations in Signals and Systems

Several interactive simulations that promote a better understanding of concepts traditionally encountered in an undergraduate Signals and Systems course are described.

All simulations use graphical control elements that allow the student to interact with the simulation. Many of the simulations use either real audio or image data so that the student can vary parameters of the processing operation and immediately hear or see the effect.

The simulations are written in MATLAB, but they also run in Octave. Since Octave is free software and runs on all major operating systems, the student need not purchase any software to run the simulations. Since no specialized computer systems or software is required the simulations are well-suited for use in an online course.

The software is modular and well-documented. The wide-variety of simulations available means that the collection forms a nice framework for development of additional simulations.

Motivation

A new textbook [1] was adopted for the Signals and Systems (EE 310) course at the University of Evansville for the 2018 fall semester. The textbook described several interactive computer simulations that were written to further enhance student learning. (A PDF version of the textbook and simulation software are available for free download [3].) The simulations are written in LabVIEW [4]. Unfortunately the students taking the course have limited access to LabVIEW within the college of engineering. (Although, free evaluation and inexpensive student versions of LabVIEW are available [4].) Additionally, students have typically not used LabVIEW prior to taking this course.

At the same time the textbook was being considered for adoption, the author was considering developing interactive simulations for a Signals and Systems course in Octave [5]. (Octave is free numerical computation software with syntax that is compatible with MATLAB [6].) The students in the Signals and Systems course have previously taken a course in MATLAB programming. Students have easy access to both Octave and MATLAB in all computers in the college of engineering. Most of the students have Octave or MATLAB installed on their personal computers.

Over the summer, the simulations in the textbook were re-written in Octave. The goal was to produce applications that behaved as similarly as possible to those on the textbook web site. There were a couple of advantages to this approach. The theory behind each simulation was described in the textbook, so additional documentation would not have to be written. Errors in the Octave simulations could be found by comparing results to the LabVIEW simulations.

The Octave simulations are available for download [2]. They may also be downloaded from the textbook site [3] along with the original LabVIEW simulations.

A Brief Description of the Simulations

The simulations can be categorized in a couple of different ways. Simulations related to continuous-time systems are shown in Tables 1, 2 and 3. The programs in Table 1 simulate several continuous-time signal operations. Those in Table 2 simulate several continuous-time systems. Those in Table 3 simulate a few audio processing operations. They use real audio input data and allow the user of the application to play the input and output audio data. This allows the user to hear the result of the operation.

Table 1: Simulation of Continuous-Time Signal Operations

File Name	**Description**
sim00_01.m	Function composition
sim01_01.m	Convolution of rectangular pulses
mod02_01.m	Convolution of exponential functions

Table 2: Simulation of Continuous-Time Systems

File Name	**Description**
mod02_02.m	Automobile suspension simulation
mod04_01.m	Oven temperature simulation
mod04_02.m	Inverted pendulum simulation

Table 3: Simulation of Continuous-Time Audio Processing Operations

File Name	**Description**
mod06_01.m	Remove sinusoidal interference from a trumpet signal
mod06_02.m	Separate two trumpet signals
mod06_03.m	Remove noise from a trumpet signal

Simulations related to discrete-time signals and systems are listed in Tables 4, 5 and 6. Those Table 4 are related to simulation of a variety of discrete-time operations. Those in Table 5 process real audio while those in Table 6 simulate image processing applications.

Descriptions of Selected Simulations

A few simulations will be described in more detail in this section. Full descriptions of all simulations can be found in [1].

Figure 1 shows the graphical user interface for a function composition application (Table 1 - sim00_01.m). The student can use the interface to create a function that consists of a sum of amplitude-scaled and time-shifted unit step and ramp components. A graph of the function is shown at the top of the window. A mathematical expression for the function is shown in the output text box immediately below the figure. The graphical control elements below the text box can be used to add or delete step and ramp components from the current function. This program

can be used in two ways. It can be used to create a graph of a given mathematical expression or, given a graph in a homework problem the student can use the program to verify that an expression is correct for the given graph.

Table 4: Discrete-Time System Simulations

File Name	Description
mod08_01.m	Frequency response from poles and zeros
mod08_06.m	De-reverberation (inverse filtering)
mod08_07.m	Noise removal via thresholding in the frequency domain
mod08_09.m	Spectra of periodic signals using the DFT
mod09_01.m	Filter design using windowing
mod09_02.m	Spectrogram of an audio signal
mod09_03.m	Spectrogram of a chirp signal
mod09_04.m	Using auto-correlation to estimate the period
mod09_05.m	Using cross-correlation to estimate time delay

Table 5: Discrete-Time Audio Processing Examples

File Name	Description
mod08_02.m	Remove sinusoidal interference from a sinusoid
mod08_03.m	Remove sinusoidal interference from a trumpet signal
mod08_04.m	Remove periodic interference from a sinusoid
mod08_05.m	Separate two trumpet signals via filtering
mod08_08.m	Separate two trumpet signals using the DFT

Table 6: Discrete-Time Image Processing Examples

File Name	Description
mod10_01.m	Lowpass filtering an image
mod10_02.m	Removing noise from an image

Figure 2 shows the graphical user interface for a simulation of an automobile suspension (Table 2 - mod02_02.m). Different parameters of the suspension can be adjusted using the controls in the upper right. The response to three types of pavement displacement is simulated. Either a curb, a pothole or wavy pavement can be selected using the radio button controls. Pavement parameters can be changed using the controls at the lower right. (The figure shows the response to a 0.3 cm deep pothole when the car is traveling at 2 m/s.)

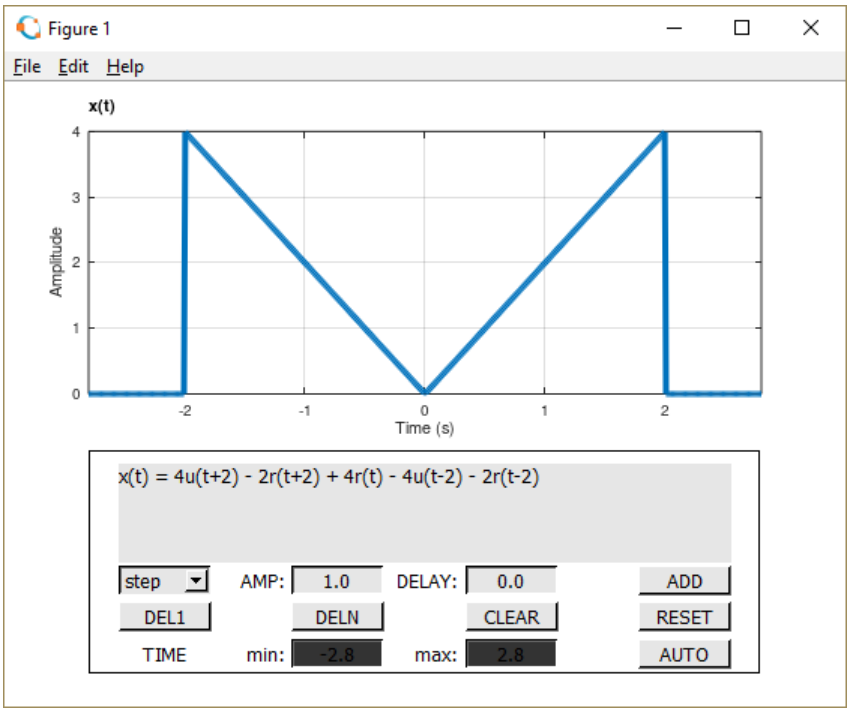

Figure 1: Function Composition (sim00_01.m)

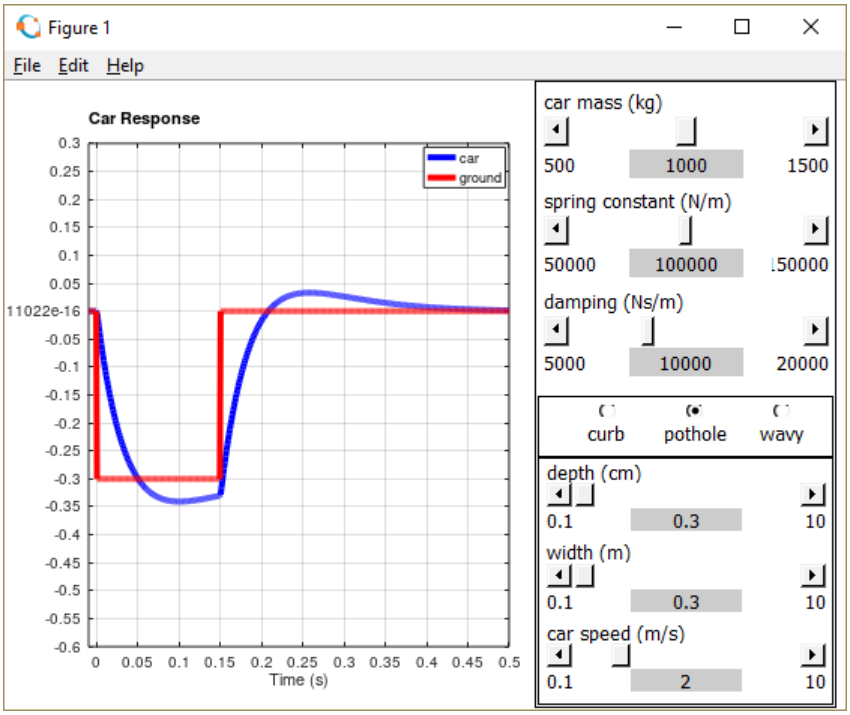

Figure 2: Simulation of Automobile Suspension

Figure 3 shows the user interface for one of the audio simulations (Table 3 - mod06_01.m). A portion of the audio waveform produced by a trumpet playing the note B is shown in the upper left. A plot of the amplitude spectrum is at top-center. The result of adding a sinusoidal interference is shown in the two middle graphs. The amplitude and frequency of the interfering

signal can be adjusted using the controls in the lower right. The original audio and the audio with added interference can be heard by pressing the buttons labeled “x” and “y”. The bottom graphs show the result of using a notch filter to remove the interference. The notch frequency and filter decay factor can be adjusted using the controls. The button labeled “z” can be used to play the filtered signal.

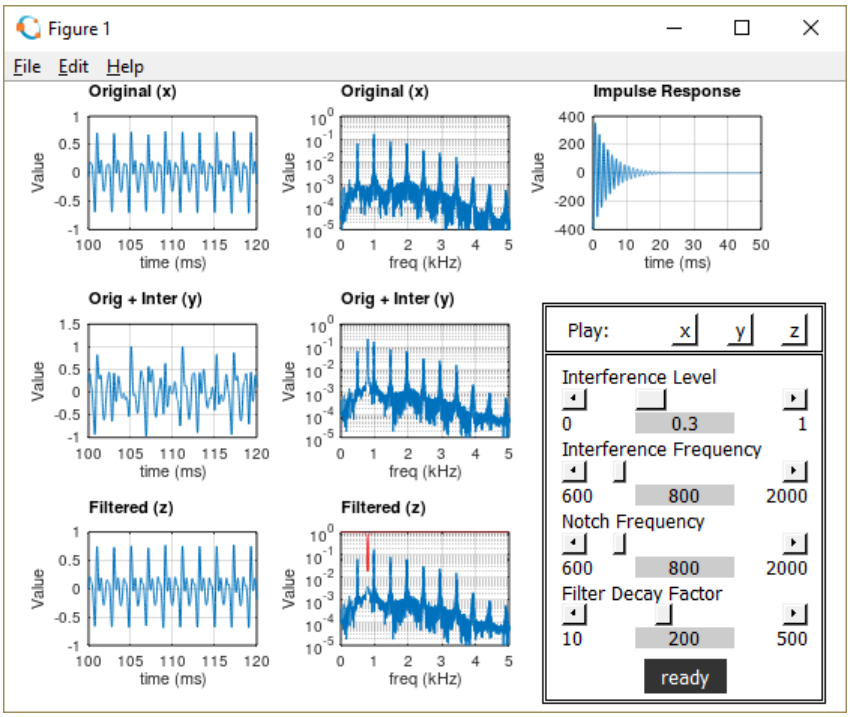

Figure 3: Removing a Sinusoidal Interference

Figure 4 shows the interface for a filter design application (Table 4 - mod08_01.m). Filter poles and zeroes can be added using the control elements at the bottom of the window. The pole-zero plot is shown in the upper left and the filter magnitude response in the upper right. The position (radius and/or angle) or a pole/zero can be changed using the controls and the pole-zero plot and frequency response are immediately update to reflect the change.

Figure 5 shows one of the image processing simulations (Table 6 - mod10_02.m). The original image and magnitude spectrum are shown at the top of the window. The image and spectrum with additive Gaussian noise are shown in the center row of images. The noise power level (variance) can be adjusted using the top-most control element to the right of the figures. The resulting image and spectrum after low-pass filtering the noisy image are shown at the bottom of the window. The cutoff frequency of the filter and window length of a Hamming window that is applied to the filter response can be adjusted. Changing the control elements results in immediate visual changes to the noisy and filtered images.

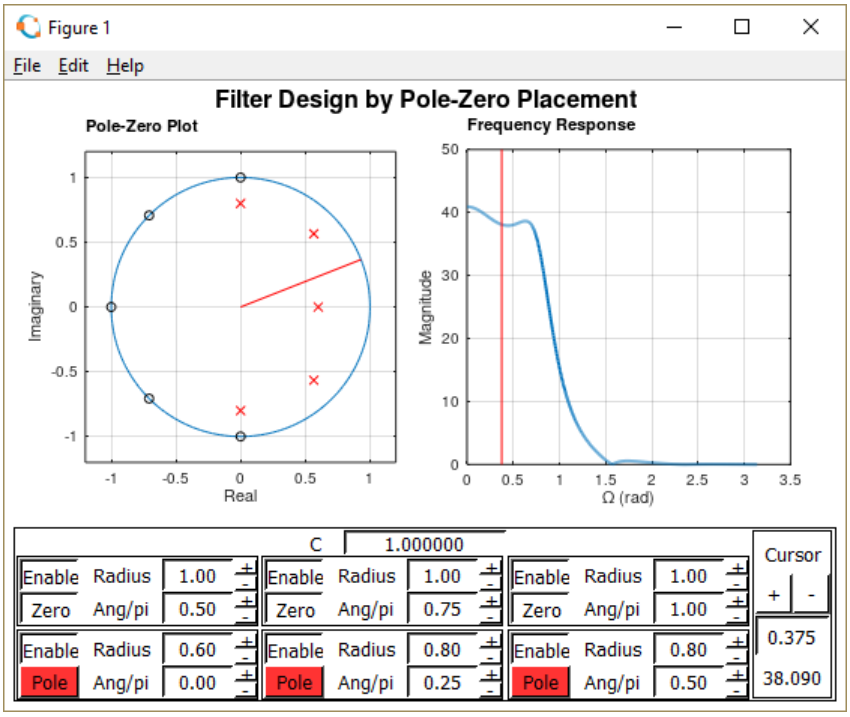

Figure 4: Filter Design

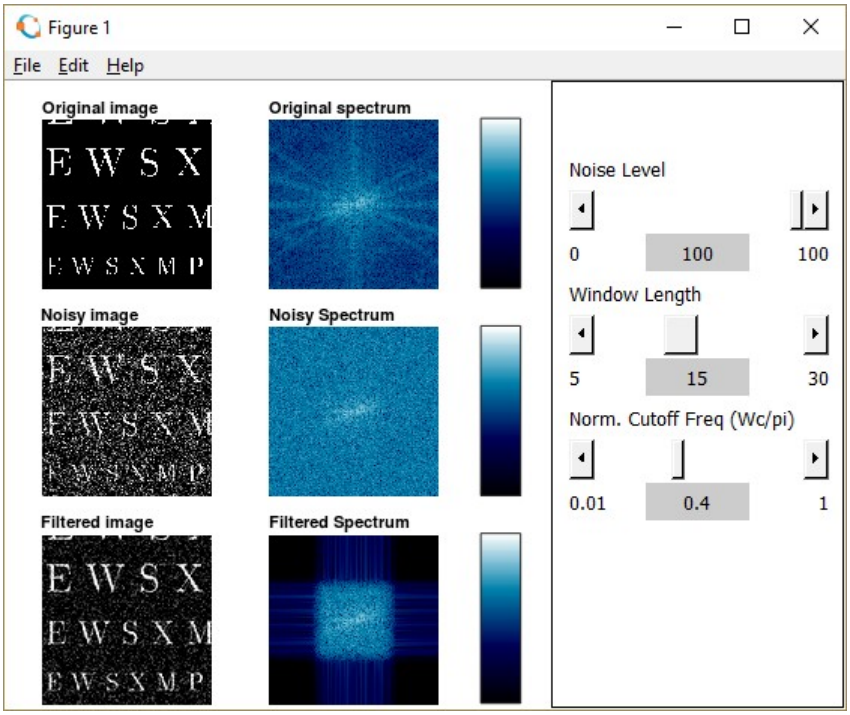

Figure 5: Removing Noise From an Image

Development Framework

Since all simulations are written in the *m*-file scripting language they can be readily modified using any text editor. Modifications can enhance or extend the available simulations.

The software is written in a modular format and can be used as a framework for the development of new simulations. Almost all of the simulations consist of just three main functions: (1) a main routine in which all default parameter values are defined, (2) a graphical user interface (GUI) function which creates all of the control elements that are presented to the user, and (3) a simulation routine that does all of the data processing and updates all output plots. A document on the web site describes how the framework can be used to develop new modules.

Conclusions

Several interactive simulation programs have been developed for demonstrating concepts in signals and systems to students. Simulations related to function composition, convolution, filtering, and filter design are available. Many of the simulations use real audio or image data so that students can hear or see the effect of a particular filtering operation. All modules are readily available for download.

The simulation programs will be used in the Signals and Systems course taught at the University of Evansville in the Fall 2019 semester. Assessment related to the effects of using the software on student learning outcomes will be presented in the future.

References

[1] F.T. Ulaby and A.E. Yagle, *Signals and Systems: Theory and Applications*, Michigan Publishing, 2018.

[2] Module Web Site [online]. Available: https://csserver.evansville.edu/~richardson/courses/EE310_LinearSystems_And_DSPI/resources/ulaby_text/octave/index.html. [Accessed: 1-January-2019].

[3] Supporting Web Site for *Signals and Systems: Theory and Applications* [1] [online]. Available: http://ss2.eecs.umich.edu. [Accessed: 1-January-2019].

[4] National Instruments LabVIEW Home Page [online]. Available: http://www.ni.com/en-us/shop/labview.html. [Accessed: 1-January-2019].

[5] GNU Octave Home Page [online]. Available: https://www.gnu.org/software/octave. [Accessed: 1-January-2019].

[6] MathWorks MATLAB Home Page [online]. Available: https://www.mathworks.com/products/matlab.html. [Accessed: 1-January-2019].