INTERACTIVE DSP EDUCATION USING MATLAB DEMOS

Umesh Rajashekar, *Student Member, IEEE* and Alan C Bovik, *Fellow, IEEE*

Laboratory for Vision Systems,
Department of Electrical and Computer Engineering
The University of Texas at Austin, Austin, TX 78712-1084 USA
{umesh,bovik}@ece.utexas.edu

Abstract

In this paper, we present a suite of interactive Digital Signal Processing (DSP) demonstration modules developed using MATLAB. Equipped with informative visualizations and a user-friendly interface, these modules are used effectively in a classroom environment for teaching DSP. Most demos use real-world signals such as audio and images so that the student can appreciate the concepts better. We focus on providing visualization tools that accentuate the intuitive aspects of DSP algorithms.

1 Introduction

Digital signal processing (DSP) deals with the analysis and manipulation of digital signals. DSP finds a wide gamut of applications in industry. Many Electrical Engineering curriculums introduce DSP in the undergraduate level while the advanced DSP course is normally covered in one or more graduate courses. The main difficulty in teaching DSP to a novice is the large number of mathematical equations that need some intuition to appreciate the underlying concept. Visualization of these abstract and non-intuitive equations can aid the student to see through and understand the math. Visualizing the concept and experimenting repeatedly using a visualization aid can be very useful to enhance the student's comprehension. MATLAB and Mathematica are capable of such visualizations. Most introductory courses involve computer-based DSP exercises to reinforce the concepts introduced to the student. However, the learning curve involved in understanding the software can distract a student from trying to appreciate the concepts for which the exercises were designed. Hence, a ready-to-use set of demonstrations illustrating the concepts that the instructor considers important can help the student to begin experimenting immediately without having to bother about the programming intricacies.

In this paper, we introduce a set of MATLAB demos that were developed for the EE 381K graduate course in DSP at the University of Texas at Austin (Spring 2000). The demos were motivated by the success of using similar visualization tools (Labview) developed for the EE 371R undergraduate course in Digital Image Processing. The demos are currently available for download from *http://pineapple.ece.utexas.edu/class/ee381k/Modules/demos.htm*

The demos explain various concepts introduced during the course such as signal representations, Fourier Series expansions, Z-transforms, filter design and non-linear filters. In these demos, we emphasize the effects of DSP algorithms on real-world signals. For example, a student can hear and visually appreciate the effects of aliasing due to signal undersampling, or the varied effects of lowpass, bandpass, or highpass filtering, the effects of using different types of filter designs, and so on.

There have been similar web based DSP educational tools developed using Java [1] and Common Gateway Interface (CGI) [2]. However, we chose MATLAB due to the following attractive features: simple programmability, impressive integrated graphical functions, highly optimized for vectorized code (suitable for DSP algorithms), availability of MATLAB for a wide variety of platforms, and extensibility to a web-based education system such as that used at UT-Austin.

Section 2 gives an overview of MATLAB. In Section 3, we describe a few demos that were developed. Conclusions are given in Section 4.

2 Overview of MATLAB

(The following is with reference to MATLAB Ver 5.2). MATLAB (MATrix LABoratory) is a high performance language for technical computing. It integrates computation, visualization and programming in an easy-to-use environment [3]. The basic data element in MATLAB is an array. Many matrix-based functions like matrix multiplication and array dot products can be executed in a fraction of time it would take to write a similar program in a scalar non-interactive language such as C or FORTRAN. MATLAB features a family of application specific solutions called toolboxes for signal processing, neural networks and wavelets to name a few. These toolboxes are a collection of functions written as M-files. The signal processing toolbox (SPtool) for example includes an interactive environment for analyzing and manipulating signals and designing filters [4]. MATLAB also has a number of easy to use plotting and graphical functions, which make MATLAB an attractive choice for developing attractive visualization applications. [3] gives an introduction to the use of MATLAB. The vectorized nature of MATLAB, the abundant collection of functions and visualization options makes it a good choice for visualizing DSP concepts. Development of an educational tool for classroom instruction needs a powerful Graphical User Interface (GUI). GUI development in MATLAB is discussed in Section 3.1. Current versions of MATLAB also provide powerful user friendly debugging tools.

3 System Overview

Using the powerful graphics and simple functionality of MATLAB, we developed a number of DSP demonstrations that can be used for a classroom teaching environment to demonstrate DSP concepts visually. Currently all demos run on the Windows NT 4.0 platform but can be easily ported onto other platforms too.

3.1 GUI development using MATLAB

MATLAB provides-GUIDE (GUI Design Environment) to develop GUIs using "drag and drop objects" such as buttons, sliders and pop-down menus to name a few. Impressive GUIs can be developed in a short time. An M-file performing a particular task for each object in the GUI is written separately using many of the in-built MATLAB functions. The GUIDE callback editor manages the actions associated with the selection of a particular object (for e.g. clicking a button) by linking an object to its respective M-file. [5] describes the GUI development in MATLAB. *Fig 1* shows a typical GUI development environment.

Once the GUI is developed, one can use the mouse and on-screen options to visualize, hear and manipulate signals (audio and images). We have used a number of signals that can be analyzed for e.g. in audio, we use male and female voices, music and standard waves (sine, chirp, square, triangle). A few demos also use images as inputs to illustrate multidimensional DSP algorithms. All audio samples used by our demo modules are .wav files that have been stored differently for ease of programming. For classroom instruction, the demos are hyperlinked from an MS-Word document. MATLAB always runs the matlab\toolbox\local\matlabrc.m on startup. We make modifications to this file so that MATLAB starts up with the respective demos when the hyperlinks are clicked. We explain a few of the demos that were developed in section 3.2.

3.2 Examples of demos

3.2.1 Signal representations

In this introductory module, we focus on the representations of an input signal in different domains i.e. in the time, frequency and the time-frequency domains. The user can select an audio sample to be viewed in any of the above domains, change its sampling frequency and the number of bits per sample, hear to the original and the modified signal. Selecting a chirp signal and sampling it appropriately can be used to effectively visualize and hear aliasing as shown in *Fig 2*. We have also developed other simple demos to illustrate the difference between continuous, discrete time and digital signals. Another demo helps a user to hear the aliasing of a sinusoid when it is undersampled at various rates.

3.2.2 Fourier Series

In this demo, a selected standard signal is decomposed into its constituent sinusoids and the user can view the frequency representation of the signal. The user also has an option of adding one frequency component at a time to see how these sinusoids add up to form the signal under consideration. This can also be used to illustrate Gibb's phenomenon wherein, a square wave is selected as the input.

3.2.3 Z-transform

This is one of the demos that demonstrates the powerful graphics capability of MATLAB. The effect of zeros and poles on the impulse and frequency response of a system can be demonstrated. The user can place poles and zeros on the Z-plane using the click of a mouse. The 3D effect (which can be rotated for any view) of the poles and zeros can be visualized along with the other options to view frequency and impulse responses too. *Fig 3* illustrates the Z-transform demo.

3.2.4 Digital to Analog reconstruction

These demos illustrate digital to analog reconstruction using Zero order hold, linear and cubic interpolations. The user can select any standard signal and sample it at variable rates and examine the reconstructed signal. Image reconstruction using similar techniques help the user to appreciate visually, the importance of the type of reconstruction used.

3.2.5 Filter design

FIR filters - The effects of various windows for the windowing technique of FIR filter design is illustrated. The user can vary the window length and examine the frequency response of these windows. A GUI for FIR filter design using windows allows the user to specify the window to be used, the filter order and other frequency specs for filter design. The designed filter can be applied to an input signal to hear the effect of the filter on the audio sample. The magnitude and phase response of the filter can be viewed too. We have similar demos for FIR filter design using frequency sampling and Remez exchange algorithm.

IIR filters - The GUI for IIR filters is similar to the one used for the FIR case except that the design is now using analog prototypes such as Butterworth, Chebyshev and Elliptical filters. The bilinear transformation is used for converting the analog filters to the digital domain. The minimum filter order is obtained using the "Auto" design mode. The pole zero plot can be also be viewed for the designed filter.

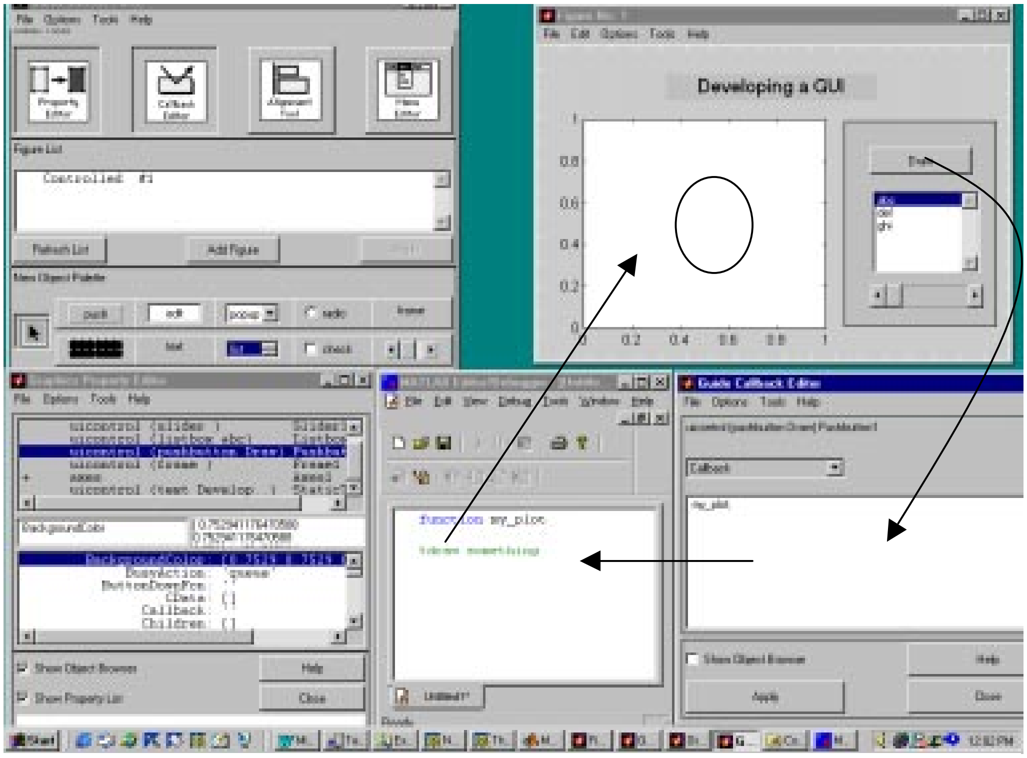
Besides the examples discussed above, we also have developed a number of other demos to explain concepts in Discrete Fourier transforms, multirate DSP, short time Fourier transforms etc. A median filter demo (for audio and images) is also developed to illustrate the superior performance of non-linear filters over linear systems.

4 Conclusions

In this paper, we discussed the development of MATLAB based demos for the visualization of DSP concepts. Currently the demos have to be downloaded and therefore need a local copy of MATLAB. MATLAB server can be used to run the demos on a server and the results can be displayed at the client end. This would obviate the need for a local copy of MATLAB at the client end. More information on the MATLAB server can be got from *http://www.mathworks.com/products/webserver/*

References

1. Cheneval Y., Balmelli L., Prandoni P., Kovacevic J. and Vetterli M., *"Interactive DSP education using Java,"* In Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, 1998. Vol 3 , Page(s): 1905 -1908
2. Rahkila M., Karjalainen M. *"An interactive DSP tutorial on the Web,"* IEEE International Conference on Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., Vol 3, 1997 , Page(s): 2253 -2256
3. ___, *"Getting started with MATLAB-Version 5,"* The Mathworks Inc May 1997
4. ___, *"MATLAB Signal Processing Toolbox,"* The Mathworks Inc Dec 1996
5. ___, *"Building GUIs with MATLAB-Version 5,"* The Mathworks Inc June 1997



```
function my_plot
%draw something
```

Fig 1:GUI development in MATLAB

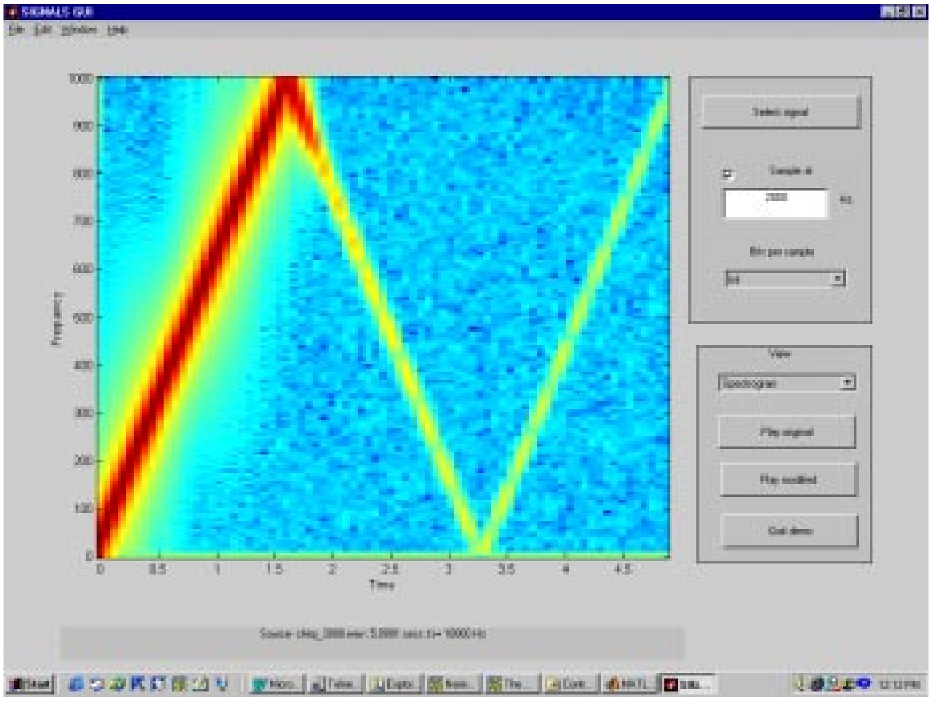


Fig 2: Signal representations (aliasing in chirp shown)

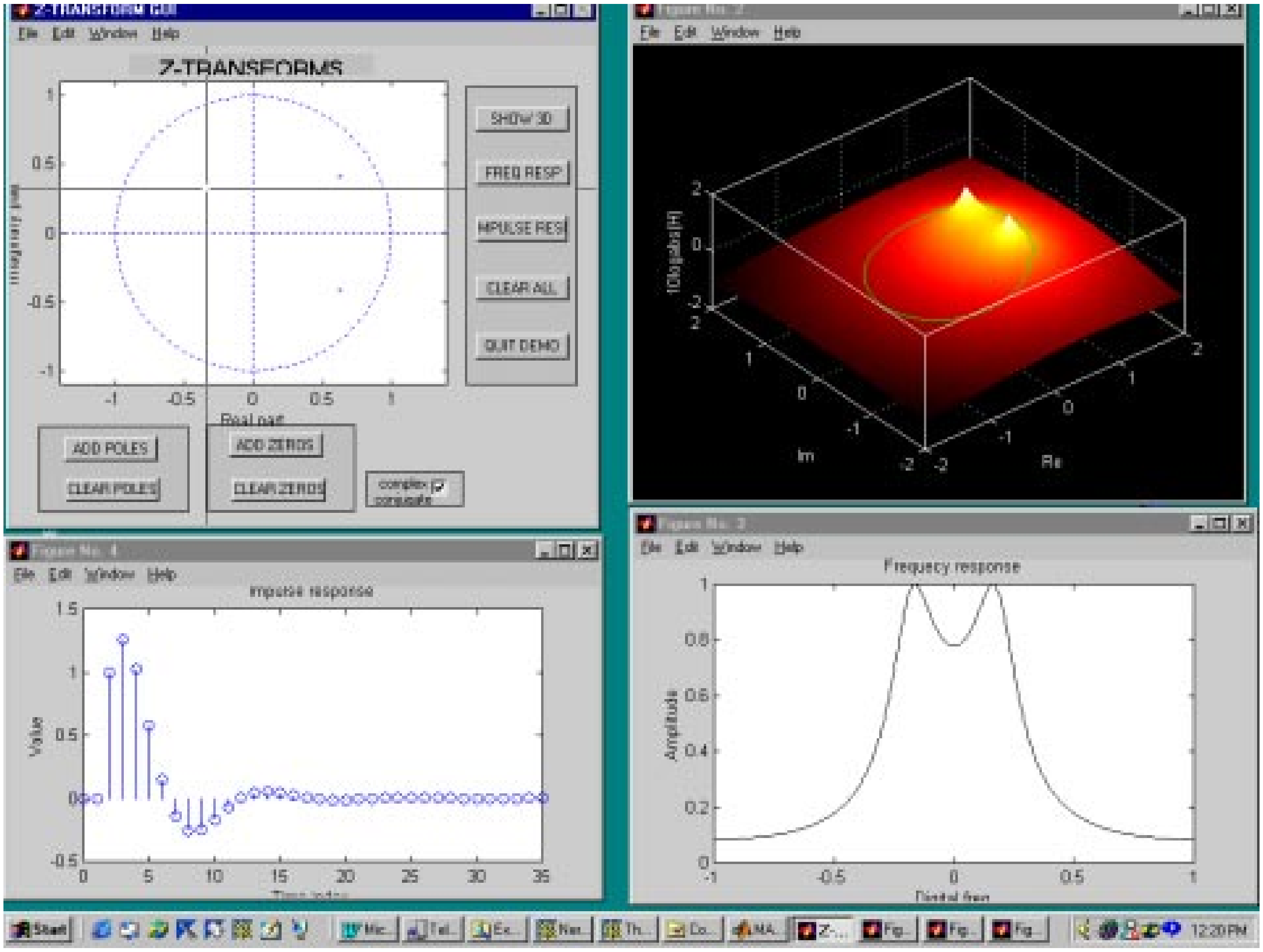


Fig 3: GUI for Z-transforms (Z-plane,3D, frequency and impulse response)