

Simulation and Analysis of Audio Signal Processor

Navneet Agrawal
Department of Electronics &
Communication Engineering
College of Technology &
Engineering, MPUAT
Udaipur, INDIA

Dhruv Jain
Department of Electronics &
Communication Engineering
College of Technology &
Engineering, MPUAT
Udaipur, INDIA

ABSTRACT

An Audio Signal Processor analyze sound signals, process them and explore the signal properties. This paper is a part of the work carried out by the authors for designing the Audio Processor in MATLAB software providing basic functions to read, play and write the sound signal along with signal processing toolbox, which includes numerous functions that work on signals. A unique feature of the paper is development of Graphical user interfacing (GUI) helping users to understand the functioning and parameters of the system project and work on it in an easy way. The paper focuses on implementation of latest audio features like jazz, pop, rock, treble, bass and myfav using filters and includes the various curves like amplitude vs frequency, amplitude v/s time etc giving information about the variation of amplitude with frequency and time of signal. The paper illustrates practical aspects of FIR filter design as well, using the Filter Design Toolbox and the Signal Processing Toolbox for this purpose. The emphasis is mostly on lowpass filters, but many of the results apply to other filter types as well. The paper concludes with time domain and frequency domain analysis of the signals.

Keywords

Component; formatting; style; styling; insert (key words).

1. INTRODUCTION

The processing of musical signals is very important in both the production and composition of music. Audio effects have become commonplace since their advent in the late 1960's. As an emerging technology, engineers were forced to rely on circuit elements to get the desired effect. Although technically effective, this level of technology was time-consuming from a production perspective and yielded a product that was both bulky and expensive. The creation and refinement of digital signal processing coupled with other important advancements [5] in the field of engineering have moved audio effects into the mainstream. By its very nature, digital signal processing has enabled the development and production of very high quality effects that are far less bulky and vastly less expensive. Real-time audio processing is a full-fledged subject in itself and these days software and hardware audio processors are parts of musical performances everywhere. Through the basic software and hardware tools, we wanted to explore this field. With background in Fourier transforms, circuits and MATLAB coding, the paper seemed to be a good mix of these various parts.

The Paper includes designing one of the most robust GUI to remove Complexity, Computing the samples of the signal on real time, Plotting real time FFT [13] of the signal that includes time domain plot, frequency domain plot (real Time, complex and imaginary), applying filters to realize the effect of sound in the music player that includes pop, jazz, rock treble, bass, myfavourite etc.

"MATLAB Analog Signal Processor" (MASP) is designed first for practical effective demonstrations, second to provide an interactive experience, and third to serve as a repository of algorithms and code. Using MASP, a lecturer can quickly illustrate concepts, and a student can gain a deeper understanding of the material. The code behind the application is open and free to be used and within this paper we present this suite of applications and review its use.

2. CHOICE OF MATLAB

There are several criteria in the development of our suite of demonstrations and applications of MASP. First, the concepts must be presented clearly with little interfering information. Second the applications should be direct, flexible, and fast. Third, sound and visuals must be used to demonstrate concepts. Fourth, the demonstrations should allow the user to explore the topic by changing parameters. Fifth, people should be able to look into the code to understand how it works. Sixth, the demonstrations should be compatible with as many computer platforms possible. And finally, the cost to be able to run the applications on your own computer should be minimal. A good overview of these and other packages in terms of engineering education can be found in. Mathematica is meant more for symbolic mathematics than creating applications, and it cannot easily produce sound. Octave, a free open-source mathematics software application, is quite compatible with MATLAB code, but it lacks much of the rich library of functions available in MATLAB. In addition there is no easy way to create graphical user interfaces (GUIs). MATLAB provides a flexible integrated programming environment that is easy to use and understand, and inexpensive for students. MATLAB is platform independent, has superior graphics handling and visualization capabilities, and has a great GUI development environment for creating applications.

3. MASP GRAPHIC USER INTERFACE

Creating GUI programmatically allows the user to explore his/her programming skills and develop a better and deep knowledge of how controls are created and their functioning is associated to them. When creating small applications it is easy to remember how program is working i.e. executing statement

by statement and control is switching .Even if there is one or more function then user knows what are the I/O parameters of a particular function but when there is a large program owing to particular application which consists of large no functions(sub programs) then it becomes difficult to remember what are the I/O parameters of each function and how control is switching, rather the problem becomes more complicated when debugging program. We have easily created the text boxes as well as command buttons that go well verse with the project as well as a colorful serve as the purpose of user friendly and user will never get bored. A typical GUI created is depicted in figure 1 at the end of the paper.

4. MASP SIGNAL AND SYSTEMS

SSUM uses and extends the programming style used in the excellent “MATLAB Auditory Demonstrations” application. Modularizing the code and keeping the GUI separate from the functionality makes SSUM much more manageable. When a new application is desired, it is quite easy to copy and reuse the functionality. The applications demonstrated in our work follows:Fourier Spectrum, shown in fig 2, enables one to look at the spectrum of a sound. As the user drags a window across the time-domain representation of a signal, demarcated by two red vertical lines, the spectrum changes. The window size and shape can be changed. It would be ideal to have the window sweep as the sound plays, but currently MATLAB cannot handle such tasks. The samples help us demonstrate how continuous signals are sampled, quantized, and reconstructed. Using fourier spectrum one can investigate the cause and effect of aliasing, the effects of quantization, and the process of making digital signals continuous using ideal low pass filtering. The fig 2 represents the time domain plot of the signal.

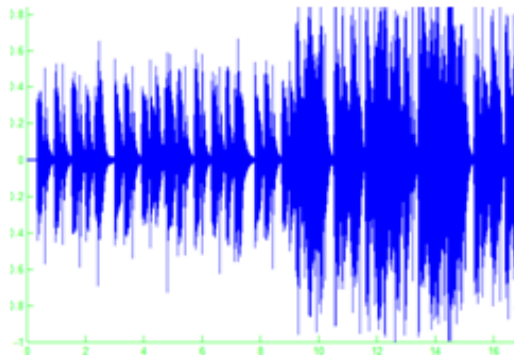


Fig.2. Time Domain Representation

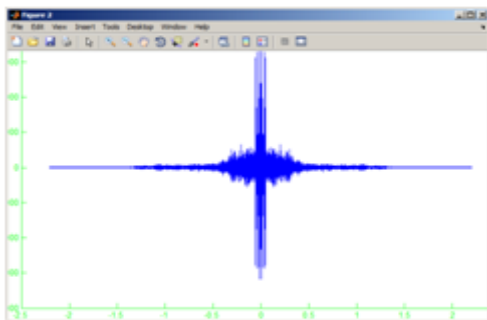


Fig3: Frequency domain (Real)

Fig 3 &4 represents the real and imaginary values of the frequency domain curve respectively. Fig 5 represent the absolute value in frequency domain of the signal (Random signal).

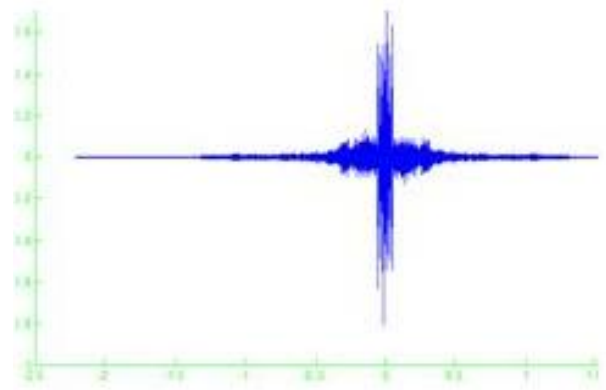


Fig4: Frequency domain (Imaginary)

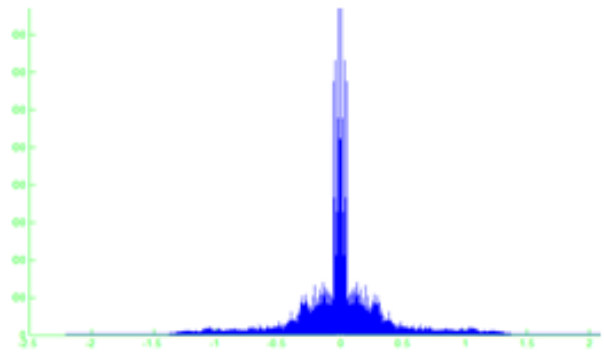


Fig5: Frequency domain (Absolute)

The bottom box in the main project shows the result of the number of samples in the track making a reconstruction back to a continuous signal. With the sliders the user can select the track on his own on which he wanted to perform processing while in text boxes the user can see the number of samples and the channels along with the effect applied on the song.

5. MY FILTER BOX

A GUI for FIR filter design using windows [13] 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. 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[1]. 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 plots 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[1] DSP, short

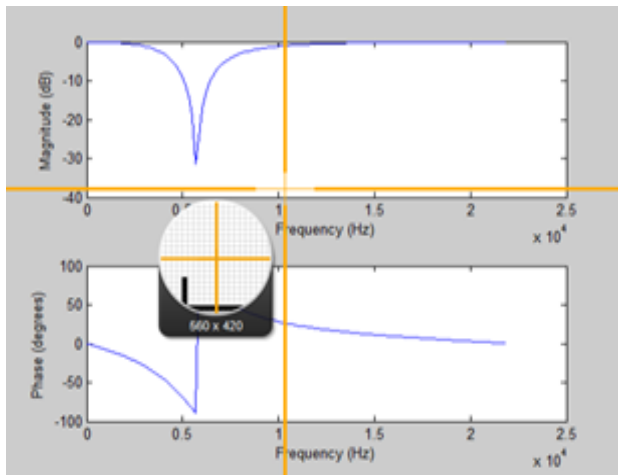
time Fourier transforms etc. A median filter demo (for audio and images) is also developed to illustrate the superior performance of non-linear filters over linear systems.

6. CONCLUSION

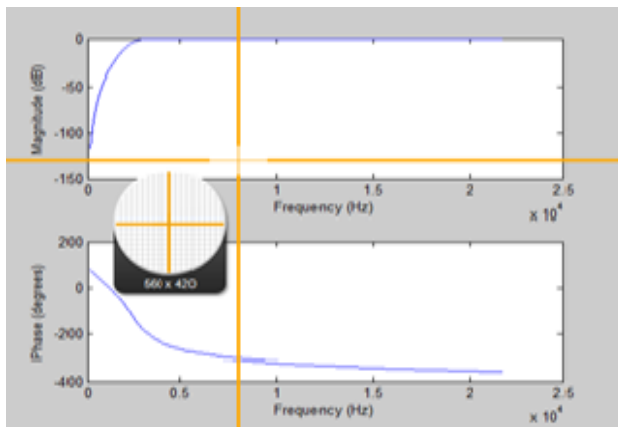
By digitizing an analog audio signal and using a DSP, an audio signal can be manipulated in an infinite amount of ways that can sound pleasing to the human ear and benefit the musician. A single DSP can do the work of hundreds of analog processors, and can do things that analog processors are unable to do, as well. It is very convenient for musicians to use DSPs for audio effects because DSPs condense all of the functions and space of analog circuits. For our work, we have successfully implemented and explored the realm of audio effects, using digital signal processing to manipulate audio signals. We have designed my filter box that could be used to apply various kinds of filter on song along with their cutoff frequency and order up to maximum of 3.

7. RESULTS

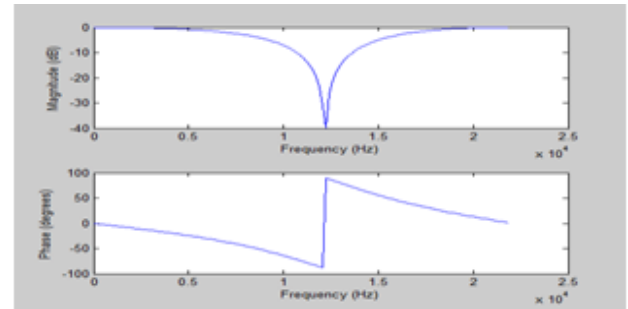
In our work we have experienced several sophisticated effects that can be applied over audio signal and thus giving effects to the signal by applying various low pass and high pass filters of different orders. The effects include jazz, pop, rock, treble, bass; myfav etc. The results are depicted as follow.



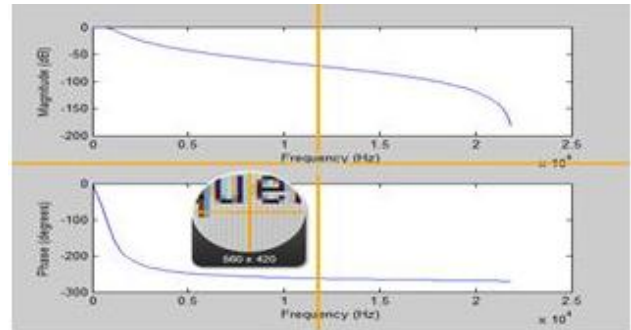
HeadPhone Effect



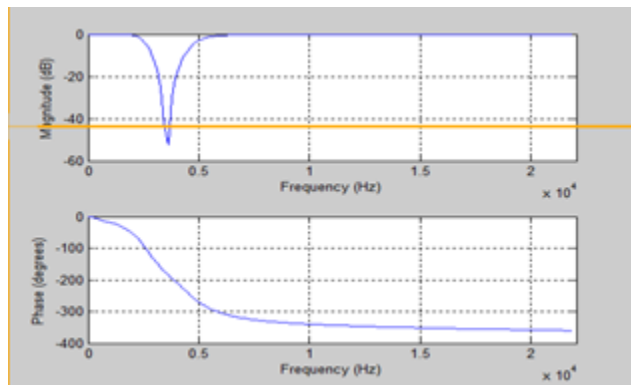
Treble Effect



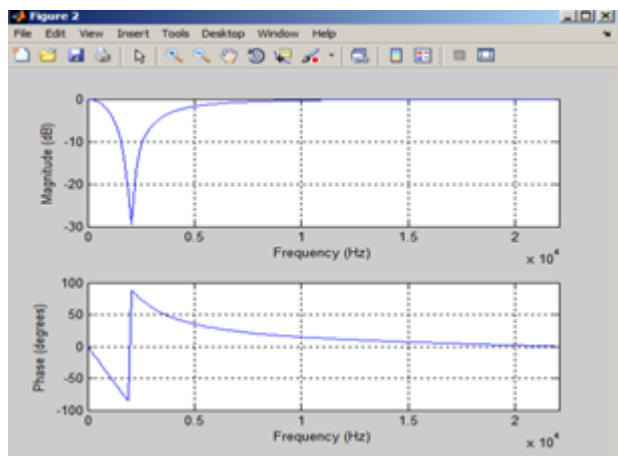
Large Hall



Bass



Myfav



Rock

8. REFERENCES

- [1] J. H. McClellan, R. Schafer, and M. A. Yoder, *DSP First: A Multimedia Approach*. New Jersey: Prentice Hall, 1998.
Signal Processing First. New Jersey: Prentice Hall, 2003.
- [2] K. Steiglitz, *A DSP Primer: with Applications to Digital Audio and Computer Music*. Menlo Park: Addison Wesley, 1996.
- [3] P. R. Cook, *Real Sound Synthesis for Interactive Applications*. Massachusetts: A. K. Peters, 2002.
- [4] U. Zölzer, Ed., *DAFx: Digital Audio Effects*. New York: Wiley, 2002.
- [5] R. J. Radke and S. Kulkarni, "An integrated MATLAB suite for introductory DSP education," in *Proc. of the First Signal Processing Education Workshop*, 2000.
- [6] M. Rahkila and M. Karjalainen, "Considerations of computer based education in acoustics and signal processing," in *Proc. of 28th ASEE/IEEE Frontiers in Education*, 1998.
- [7] J. McCartney, "Supercollider: A new real-time sound synthesis language," in *Proc. of the Int. Computer Music Conference*,
- [8] M. Puckette, "Pure data," in *Proc. of the Int. Computer Music Conference*, 1996.
- [9] M. Nagrial, "Education and training in engineering software and applications," in *Int. Conference on Engineering Education*, 2002
- [10] M. Cooke, H. Parker, G. J. Brown, and S. N. Wrigley, "The interactive auditory demonstrations project," in *Eurospeech Conference*, 1999.
- [11] F. R. Moore, "An introduction to the mathematics of digital signal processing: Part I: Algebra, trigonometry, and the most beautiful formula in mathematics," *Computer Music Journal*, vol. 2, no. 1, 1978.
- [12] J. Harvey, "Mortuos Plango, Vivos Voco: A realization at IRCAM," *Computer Music Journal*, vol. 5, no. 2, 1981.
- [13] F. J. Harris, "On the use of windows for harmonic analysis with the discrete fourier transform," in *Proc. of the IEEE*, vol. 66, no. 1, 1978.
- [14] A. Lee, "Designing a relevant lab for introductory signals and systems," in *Proc. of the First Signal Processing Education Workshop*, 2000.
- [15] J. Koumi, "Designing for learning—effectiveness with efficiency," in *Effective Screenwriting for Educational Television*, R. Hoey, Ed. U.K.: Kogan Page Ltd., 1994, pp. 230–239.
- [16] B. L. Sturm, *SSUM: Signal and Systems Using MATLAB; Creating an Effective Application for Teaching Media Signal Processing to Artists and Engineers*, 2004, (M.S. Project) University of California, Santa Barbara, Graduate Program in Media Arts and Technology, USA.