Digital Filter Graphical User Interface

Tony Finn
University of Southern Maine

Follow this and additional works at: https://digitalcommons.usm.maine.edu/thinking_matters

Recommended Citation
https://digitalcommons.usm.maine.edu/thinking_matters/135

This Poster Session is brought to you for free and open access by the Student Scholarship at USM Digital Commons. It has been accepted for inclusion in Thinking Matters Symposium by an authorized administrator of USM Digital Commons. For more information, please contact jessica.c.hovey@maine.edu.
**Problem Statements**

- Digital FIR (finite impulse response) filter design requires tedious computations, with each requiring truncation of an impulse response (seen in Figure 1.)
- In order to obtain the desired effects from a filter, one may need to try multiple filters, so many computations would be necessary.

Therefore the desire to simplify the digital filter design process is necessary to provide users an easier, more intuitive method for design.

**Objectives**

- This project aims to create an audio FIR filter design graphical user interface (GUI) in Wolfram Mathematica version 11.2, motivation and ideas derived from the user interface included in the equalizer in Logic Pro X, seen in Figure 2.
- The two design types to be included in this user interface are the Equiripple and Least Squares method.
- The Equiripple algorithm will take transition regions and amplitudes as inputs and will meet those specifications by compromising the magnitude of the ripples in the stop bands and passbands.
- The Least Squares method will take cutoff frequencies and amplitudes as inputs and will meet these specifications by compromising the width of the transition regions and the magnitude of the initial ripples around the transition regions.
- Have the ability to monitor the frequency spectrum of the audio object in which manipulation is desired.

**Results**

Illustrated in Figure 3 is the final design of the user interface, one will find buttons to change design type and filter type as well as clickable buttons to give the user feedback or an output.
- **Play Original Audio**: emits the input audio as is; unfiltered.
- **Play Filtered Audio**: emits the input audio with the designed filter applied.
- **Return Filtered Audio**: returns the filtered audio.
- **Print Filter**: returns the filter specifications.

**Applications**

- Mastering and separating audio in music such that doubled tracks won’t overlap the same frequency bands.
- Turning mono audio into stereo to add a dynamic feel such as mimicking a helicopter flying from left to right.
- Filtering out certain frequencies which may be amplified due to the undesirable frequency response of microphones, speakers, or room acoustics.

**Constraints**

Due to lower resolution in the lower octaves, filters with desired manipulation around 1kHz and below may not be able to guarantee a result satisfactory results as the resolution is low, on a logarithmic scale, as a filter with the same length would in the higher octaves. An illustration of this is below in Figures 4 and 5.

**Acknowledgements**

Special thanks to Dr. Mariusz Jankowski for advising and guiding the project.

References

* Wolfram Mathematica Documentation and Signals and Systems by Alan V. Oppenheim