

Homework 2

Introduction to Multimedia

2024/04/02

Create your own FIR filters to filter audio signal

- Input:
 - HW2_Mix.wav
- Goal
 - Using 3 filters (Low-pass, High-pass, Band-pass) to separate three audio signals from the given audio file
 - Add other effects to the filtered signal

Demo

- HW2_Mix.wav
- Low_pass_XXX.wav
- High_pass_XXX.wav
- Band_pass_XXX.wav

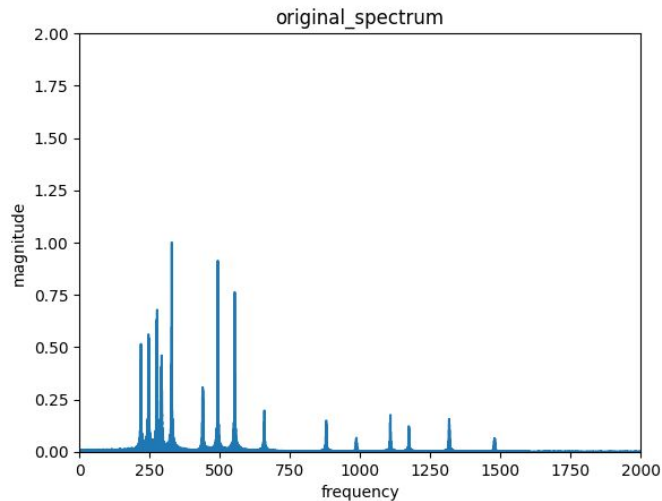


Grading

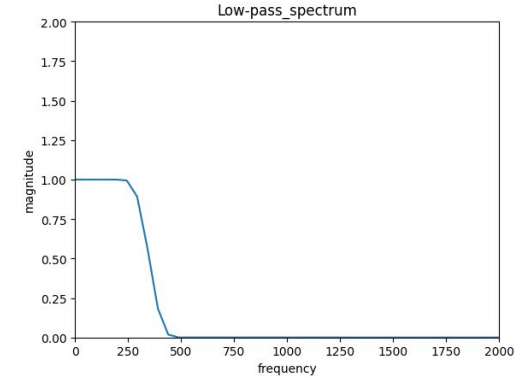
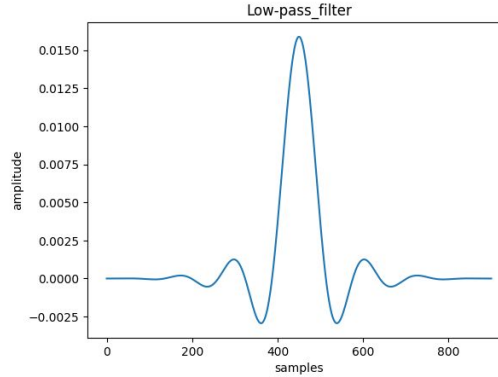
1. FFT & Plot Spectrum (15%)
2. Filter Design (25%)
3. Convolution (15%)
4. Sample rate (10%)
5. Echo (15%)
6. Report (20%)

FFT & Plot Spectrum (15%)

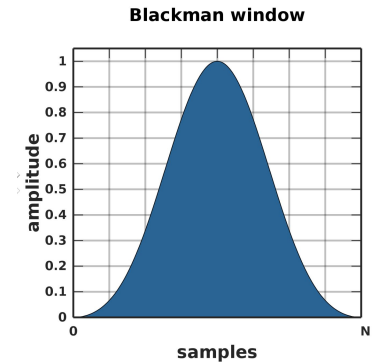
- Finish `plot_spectrum()`
- Here's the things you need to do:
 1. Implement FFT
 2. Plot the spectrum (remember to normalize the magnitude to $[0, 1]$)
- You can use `numpy.fft` and `numpy.fft.fftfreq`



Filter Design (25%)

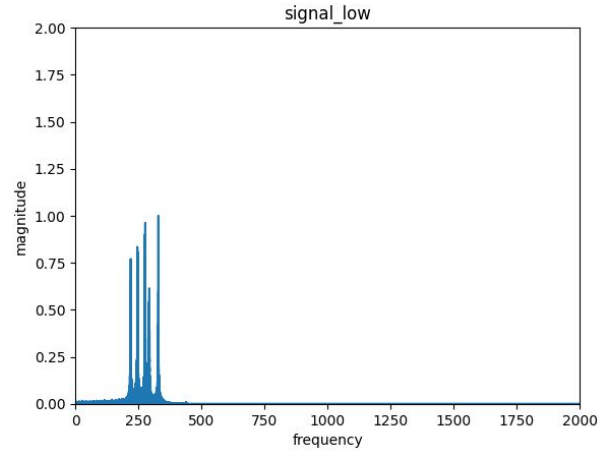


- Finish filter() & Finish parameters setting in main()
- Here's the things you need to do:
 1. Implement 3 filters (Low-pass, High-pass, Band-pass)
 2. Implement window function (Blackman)
 3. Show the filter **shape** and filter **spectrum**



Convolution (15%)

- Finish `convolve()`
- Here's the things you need to do:
 1. Do 1D Convolution on the original signal (zero padding)
- Do **not** use `np.convolve()`



Sample rate (10%)

- Finish `down_sampling()`
- Here's the things you need to do:
 1. Down sample each filtered signal to 2000 Hz (original is 44100Hz)

Echo (15%)

- Finish echo()
- Here's the things you need to do:
 1. Creates a **single** echo effect on the signal that passes through the **low-pass filter**, which means it adds the original signal to a delayed and attenuated version of itself.
 2. Creates a **multiple** echo effect, similar as above, but each echo itself is also echoed.



Report (20%)

Your report should cover the following aspects:

1. Describe how you implemented the filter and convolutions to separate the mixed song. And how did you determine the filter size and cut-off frequency? (5%)
2. Compare the spectrum and shape of the filters.(5%)
3. Compare the differences between the signals before and after reducing the sampling rates.(5%)
4. How did you implement one/multiple fold echo?(5%)

Submission

- Submit HW2_yourID_yourName.zip to eeclass before the deadline. File structure as follows:
- **Deadline: 2024/04/16**

✓ HW2_YOURID_YOURNAME

✓ output

- 🔊 Band_pass_XXX_XXX_2khz.wav
- 🔊 Band_pass_XXX_XXX.wav
- 🔊 echo_multiple.wav
- 🔊 echo_one.wav
- 🔊 High_pass_XXX_2khz.wav
- 🔊 High_pass_XXX.wav
- 🔊 Low_pass_XXX_2khz.wav
- 🔊 Low_pass_XXX.wav

✓ src

- 🔊 HW2_Mix.wav
- 📄 HW2_yourID_yourName.ipynb

Global Grading Policy

- Call built-in function: 0% for that part
 - Implementation such as: *filter design*, *convolution1D*, *echo*, *down sample* should implement **by yourself**.
- Late submission: -20% per day
- Do not follow the required file structure: -5%

Reference

- **Course Material:**
 1. FFT transform : Unit4 p23~p34
 2. FIR filter design : Unit4 p68~p74 (Blackman windowing function use p36, not p75)
 3. Echo : Unit4 p65