EE3025 Assignment-1

Sharath Chandra - ES18BTECH11016

Download all codes from here

https://github.com/tnfssc/IITH_EE3025_H12021/tree/master/assignment-1

And TeX codes from here

https://github.com/tnfssc/IITH_EE3025_H12021/tree/master/assignment-1

1 Problem

Modify the following code given in problem 2.3 with different input parameters to get the best possible output.

```
import soundfile as sf
from scipy import signal

#read .wav file
input_signal,fs = sf.read('Sound_Noise.wav')

#sampling frequency of Input signal
sampl_freq=fs

#order of the filter
order = 3

#cutoff frequency 4kHz
cutoff_freq=4000.0

#digital frequency
Wn=2*cutoff_freq/sampl_freq

# b and a are numerator and denominator
    polynomials respectively
b, a = signal.butter(order,Wn,'low')

#filter the input signal with butterworth filter
output_signal = signal.filtfilt(b, a, input_signal)
#output_signal = signal.lfilter(b, a, input_signal)

#write the output signal into .wav file
sf.write('Sound_With_ReducedNoise.wav',
    output_signal, fs)
```

2 Solution

The parameters that can be modified here are:

- Order of the filter
- Cutoff frequency
- Cascade filtering

2.1 *Cutoff frequency*

To get a good cutoff frequency, we can observe the spectrogram from Problem 2.2. Here, the synthesizer key note frequencies are between 440Hz and 5.1KHz. So, let the midpoint be the cutoff frequency i.e, cutoff frequency = (440Hz + 5100Hz)/2 = 2770Hz.

2.2 *Order*

The main disadvantage of the Butterworth filter is that it achieves the pass band flatness at the expense of a wide transition band as the filter changes from the pass band to the stop band. We could fix it by increasing the order. However, such high orders can produce miniscule values hard to work with while simulating. So, let the order be 4.

2.3 *Cascade filtering*

Since the order is just 4 now, there will be a lot of noise even after filtering. This is because the transition region is too wide. To make the transition steeper, we could increase order, but we can't. So, we can cascade the filter to get the values after the cutoff frequency closer to zero. The more we cascade, the steeper the transition region.

3 Results

To get a better quantitative idea about how much noise was filtered, we can calculate the integral of the output signal and see the difference between filtered noises.

	gvv-filter	**my-filter**
Cutoff	4000 Hz	2770 Hz
Order	3	4 (cascaded 20 times)
Integral 0 to cutoff	$1.67 * 10^7$	$1.48 * 10^7$
Integral cutoff+	$2.2 * 10^5$	$1.33 * 10^5$
Ratio (0-cutoff/cutoff+)	0.0134	0.0089

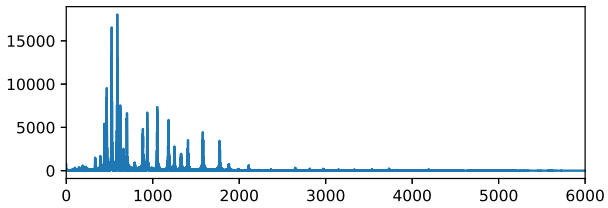

Fig. 0. Sound With ReducedNoise

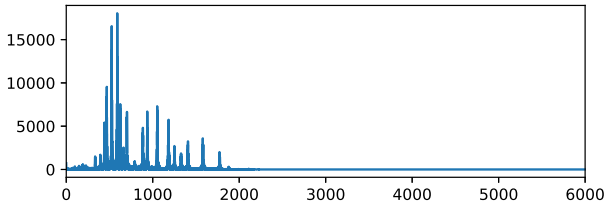

Fig. 0. Sound Low Noise