Test task to apply for internship program

Prior knowledge required.

1. C programming experience
2. Mathematics background
3. Digital Signal Processing basics
4. Common sense

Problem description

When singer sings into microphone sometimes it produces sibilants which sound like a whistling. Design digital system which will eliminate these unwanted sounds. Select DSP and implement core function in C which will process signal sample by sample.

Delivery

1. Explanation how you are going to solve the problem. Why you select this DSP core? How you select sampling rate?
2. Provide with a C program which perform required filtering

The aim of the work is to develop a set of digital filters, which must decrease a sibilant noises from a given signal.

Sibilant noise appears in vocals due to different cases: bad choice of microphone and initial recording parameters or specific anatomy of singer's mouth. This sounds are "s", "z", "ch", "sh" which is called the "the ess sounds" in music. Due to my research I find, that this sounds placed in band of 2-10 kHz. It's impossible to develop one universal filter with constant parameters, which decreases this sounds, because the different peoples have different voices frequencies (woman's voice is higher than man's) and different manner of singing (bass, baritone, soprano etc.). Also the sounds are differ from each other in frequencies, and when we decrease one sound, other sound might be stay.

I'm going solving this problem in the next way. The sibilant sound appears at high frequency. Therefore, we can attenuate it with the help of low-pass filter. But in this case, attenuation will affect for all high-frequency band, that not satisfying initial recommendations. We need to find sibilant sound in spectrum and attenuate only needed bandwidth. This is possible with using band-stop filter. But regulation of attenuation in this case became difficult, because every attenuation attempt needed re-calculation of the digital filter and normal pronouncement of this sounds attenuates too, that is also unsatisfied. The better way is split signal on sibilant noise signal and a signal without sibilant noise. We attenuate only narrow sibilant noise signal. After that we unite both signals together. Also I read lots of forums, read technical literature (namely "Audio Effects. Theory, Implementation and Application" by Joshua D. Reiss and Andrew P. McPherson) and watch on YouTube how other peoples solve this task. And after that I found wright way in solving this task.

For decreasing of this sounds are need to develop a special filter which is called a De-Esser. De-Esser works in a following way: at first needed to find the frequency

band in which wanted friquencies is located. As higher mentioned, the needed band is in range of 2-10 kHz. We split input signal:

- At first branch we need to cut-off the finding frequency range from the input signal. We doing this by adding a band-stop filter with the chosen lower and higher cut-off frequencies in given range.
- At second branch we need to add a band-pass filter with the lower and higher frequencies in a given range, with the aim to get a sibilant sound signal and a parallel connected compressor, which reduce the dynamic range of the given signal.

After that we re-united this two branches together by addition and get the output signal.

Scheme is shown at the figure below:

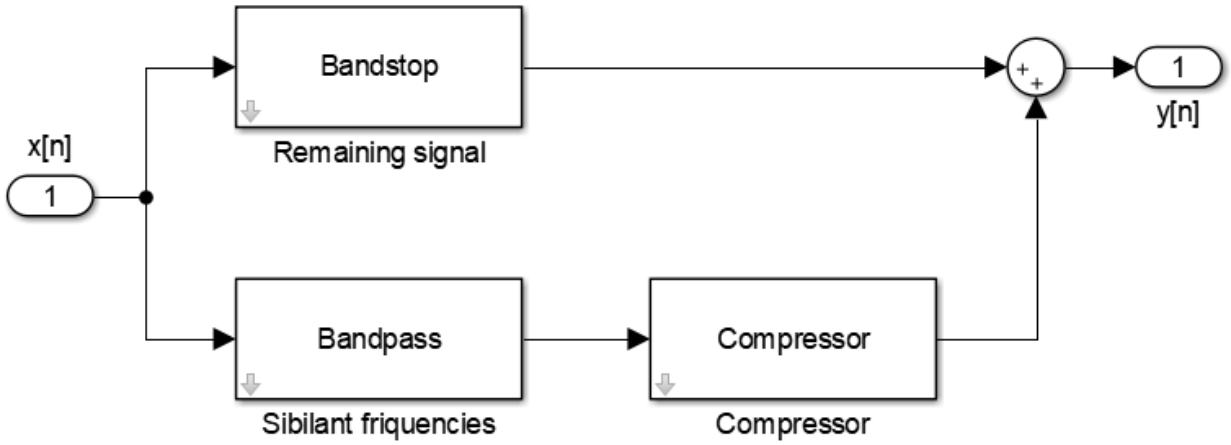

Figure 1 – Split-band De-Essing scheme

Now let's chose a sampling rate. Humans voice placed in frequency range approximately of 85 – 400 Hz. But it also care a lot of overtones (additional higher frequencies), through witch we can discern different voices. Therefore better choice is select higher sampling rate, with the aim not to lose lots of voice information. Audio standards consist a set of recommended sampling rates for audio recording. I choose a sampling rate of 44100 Hz, because this sampling rate fully covered all possible

frequencies, which human ear can hear. Also this sampling rate have good frequency margin and provided by a standard recording equipment.

Dynamic range compression (or just compression) is concerned with mapping the perceived dynamic range of an audio signal to a smaller perceived range. Dynamic range compressors achieve this goal by reducing the high signal levels while leaving the quieter parts untreated.

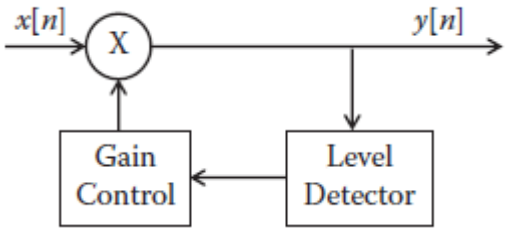

Figure 2 – Compressor scheme

Compressor has a lot of parameters:

- Threshold – defines the level above witch compressor activated.
- Ratio – defines input/output ratio for signals overshooting the threshold level
- Attack and release – set the react time of compressor

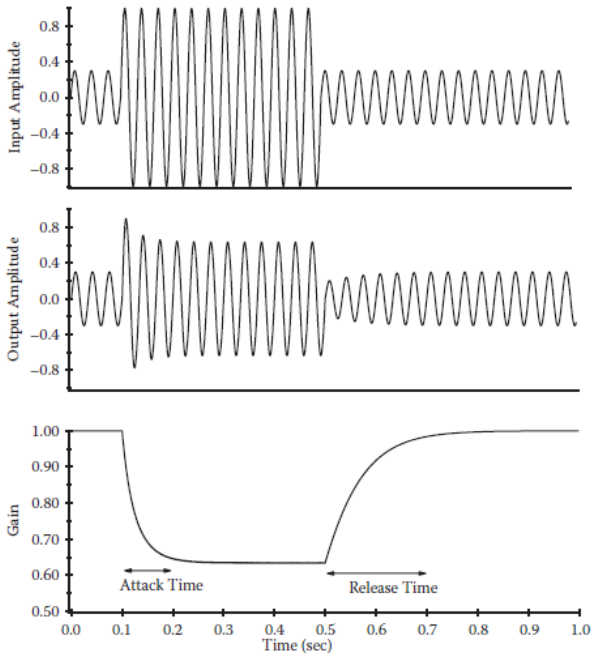

Figure 3 – Compression of a signal

Let's calculate a FIR Band-pass and Band-stop filters.

Now write recommendations for filters.

FIR-filter describes of the next expression

$y(m)=\sum_{n=0}^{N-1}h(n)x(m-n)$, where h(n) – filter coefficients, x(m) – input signal, y(m) – output signal.

1. Band-pass filter
   - Bandwidth 3 - 5 kHz
   - Transient band (f_t) 1 kHz
   - Stop bands 0 - 2 kHz and 6 – 22 kHz
   - Attenuation 70 dB
   - Sampling rate (f_s) 44.1 kHz

   Use weight method to calculate filter coefficients:

   First of all find the impulse characteristics of standard frequency-selectable filter.

   For band-pass filter formula looks like:

   $$h_D(n)=2f_2\frac{\sin(n\omega_2)}{n\omega_2}-2f_1\frac{\sin(n\omega_1)}{n\omega_1};\qquad h_D(0)=2(f_2-f_1)$$

   For our purposes we select a Blackman weight function, which satisfy our recommendations:

   $$w(n)=0.42+0.5\cos\left(\frac{2\pi n}{N-1}\right)+0.08\cos\left(\frac{4\pi n}{N-1}\right).$$

   Find length of filter (N) according to Blackman function:

   $$\Delta f=\frac{5.5}{N}=\frac{f_t}{f_c}\Rightarrow\Delta f=0.02267\Rightarrow N=\frac{5.5}{\Delta f}=243;$$

   $-121\le n\le 121.$

Predicting the smooth effect of filter characteristic, caused by weight function:

$$f_{c1}^{/} = f_{c1} - \frac{f_t}{2} = 3 - 0.5 = 2.5kHz \Rightarrow \frac{2.5}{44.1} = 0.05668$$

$$f_{c2}^{/} = f_{c2} + \frac{f_t}{2} = 5 + 0.5 = 5.5kHz \Rightarrow \frac{5.5}{44.1} = 0.1247$$

Function $h(n)$ is symmetrical, calculates only values of $h(0)$, $h(1)$, …, $h(121)$

Find a $h(n)$ value:

$$h(n) = h_D(n) \cdot w(n)$$

2. Band-stop filter

- Bandwidth 0 - 2 kHz and 6 – 22 kHz
- Transient band 1 kHz
- Stop band 3 - 5 kHz
- Attenuation 70 dB
- Sampling rate 44.1 kHz

Use weight method to calculate filter coefficients:

At first find the impulse characteristics of standard frequency-selectable filter.

For band-pass filter formula looks like:

$$h_D(n) = 2f_1 \frac{\sin(n\omega_1)}{n\omega_1} - 2f_2 \frac{\sin(n\omega_2)}{n\omega_2}; \quad h_D(0) = 1 - 2(f_2 - f_1)$$

For our purposes we select a Blackman weight function, which satisfy our recommendations

$$w(n) = 0.42 + 0.5\cos\left(\frac{2\pi n}{N-1}\right) + 0.08\cos\left(\frac{4\pi n}{N-1}\right).$$

Find length of filter (N) according to Blackman function:

$$\Delta f = \frac{5.5}{N} = \frac{f_t}{f_c} \Rightarrow \Delta f = 0.02267 \Rightarrow N = \frac{5.5}{\Delta f} = 243;$$

$$-121 \le n \le 121.$$

Predicting the smooth effect of filter characteristic, caused by weight function:

$$f_{c1}^{/} = f_{c1} - \frac{\Delta f}{f_s} = 3 - 0.5 = 2.5kHz \Rightarrow \frac{2.5}{44.1} = 0.05668$$

$$f_{c2}^{/} = f_{c2} + \frac{\Delta f}{f_s} = 5 + 0.5 = 5.5kHz \Rightarrow \frac{5.5}{44.1} = 0.1247$$

Function $h(n)$ is symmetrical, calculates only values of $h(0)$, $h(1)$, …, $h(121)$

Find a $h(n)$ value:

$$h(n) = h_D(n) \cdot w(n)$$

Digital signal processing can be computed on any processor, because It's a set of mathematical methods, which can be implemented in different form. The main question is how fast calculations will be completed. A lot signal processing software can be run on x86 architecture. For embedded devices often used digital signal processors, which have a set of special instructions, which help calculate advanced expressions in one cycle, increasing computing abilities even at low clock frequency.

My method may have big disadvantage: filter length is very long. Methods of decreasing a filter length:

- increasing transient bond
- re-calculate filter length according to Kaiser method instead of Blackman.
- decrease sampling rate

Also in programming code I try to describe filtering in common sense. Programming code doesn't changed a lot between PC or DSP. All advantages covered by compiler optimization on DSP and don't visible for a high level programming language. Nowadays PC processors have big instruction set and compute floating as well as DSP. Big disadvantage of PC x86 architecture is absence of circular buffers, that slow down real time tasks by numerous comparisons. Instead of that digital signal processors have hardware circular buffer, that help processor compute real-time tasks.

Reading data from WAV-file, filtration and writing new WAV-file

WAV is a audio format, which have raw values of recorded sound.

Structure of WAV-format is next:

1. “RIFF” descriptor: describe the format
2. “fmt” descriptor describes audio parameters such as number of channels, sampling rate, bits per sample etc..
3. “data”

My program reads wav-file (name of file passed as a command-line parameter), filtered wanted band of frequency and writes filtered data into a new wav-file. I save a header of file, which contains a data, which are equal for both wav-files.

Created by Horielov Oleksii