Lecture 3: Audio Filter Design

Introduction

This lecture gave an overview of methods used to design digital filters, what the necessary steps are and what each of the methods can be used for. As the lecture was quite extensive, in this diary we will be mainly focusing on FIR filter design as I found that part of the lecture the most interesting. In the last section we will go over a coding example that creates a low-pass filter using a sinc filter combined with a hamming window.

FIR Filter design overview

Motivation

Firstly, when designing a filter we need to decide what our requirements are and what the use case is going to be. By doing this we can decide on the design approach accordingly as basic audio filters can be designed easily in closed form, while more advanced filters will naturally require a more complex approach such as target response approximation which requires us to solve an optimization problem. These more complex applications include sound system equalization, room correction, modelling digital music instruments, etc. Another important step of filter design is to take properties of the human hearing into account since our frequency resolution is nonlinear, i.e. we are better at perceiving low frequencies, than high.

Properties

Main two properties we look at when designing are the magnitude and the phase, both of which are extracted from frequency response given by the following equation:

$$H(e^{j\omega}) = \frac{Y(e^{j\omega})}{X(e^{j\omega})}.$$

From here magnitude response is given as absolute value of the frequency response:

$$|H(e^{j\omega})|$$

and phase response as:

$$\theta(\omega) = arg\{H(e^{j\omega})\}.$$

Other important properties include phase delay $P(\omega) = -\frac{\theta(\omega)}{\omega}$ which gives us time delay of each sinusoidal component, and group delay $D(\omega) = -\frac{d}{d\omega}\theta(\omega)$ which gives the time delay of the amplitude envelope of a sinusoid at frequency ω. Interestingly, when the phase function is linear the group delay is equivalent to phase delay.

Design

When designing a filter our goal is to approximate the target response using the designed filter. In order to achieve this we want to minimize the following error function:

$$E(e^{j\theta}) = H(e^{j\theta}) - \hat{H}(e^{j\theta})$$

where E represents the error while H and $\hat{H}$ represent the target and the approximation, respectively. This problem can then be interpreted as minimizing the euclidean (L_2) norm:

$$E_2(e^{j\theta}) = ||H(e^{j\theta}) - \hat{H}(e^{j\theta})||_2^2.$$

If h is our desired response, F is the DFT (Discrete Fourier Transform) matrix and ϵ is the frequency response error the minimization problem can be formulated as:

$$min||\epsilon||^2 = min||\mathrm{F}\hat{h} - h||^2.$$

In order to find the minimum of the function we want to find such $\hat{h}$ so that the derivative (gradient) of the function is zero:

$$\frac{\delta}{\delta h}(\mathrm{F}\hat{h} - h)(\mathrm{F}\hat{h} - h)^T = 0$$

after simplifying:

$$\hat{h} = (\mathrm{F}^{\mathrm{T}}\mathrm{F})^{-1}\mathrm{F}^{\mathrm{T}}h^{\mathrm{T}}.$$

Auditory Adjustments

As previously mentioned, it is important that we take into account properties of human hearing. In order to achieve this there are various approaches such as auditory smoothing where we average the spectrum with a sliding window, weighting functions which suppress less important frequencies or frequency warping which expands the important frequency range whilst simultaneously shrinking the less important range.

Low-pass filter using sinc filter with a Hamming window

The sinc function is defined as:

$$sinc(x) = \frac{sin(\pi x)}{\pi x}.$$

In order to represent it digitally we sample it and create an impulse response sequence given by:

$$h[n] = 2f_c sinc(2f_c n).$$

However, the filter has an issue that its delay is infinite. In order to make its use possible, we need to make it finite. In order to achieve that we apply a windowing function which gets rid of the ripple at the filter edges.

We use a hamming window given by the following equation:

$$w(n) = 0.54 - 0.46cos(\frac{2\pi n}{M-1})$$

Ultimately, we create our low-pass filter by combining (multiplying) the two.

I thought it would be particularly interesting to filter a Weierstrass function[1] as it is a real-value function continuous everywhere, but differentiable nowhere given by the following formula:

$$f(x) = \sum_{n=0}^{\infty} a^n cos(b^n \pi x)$$

where in our case $a = \frac{1}{2}$ and $b = 3$ as it needs to satisfy constraints that $0 < a < 1$, b has to be a positive integer and $ab > 1 + \frac{3}{2}\pi$.

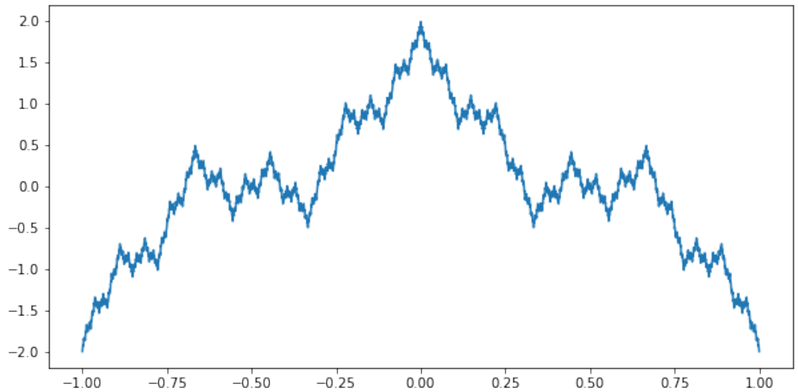

Figure 1: Original Weierstrass function.

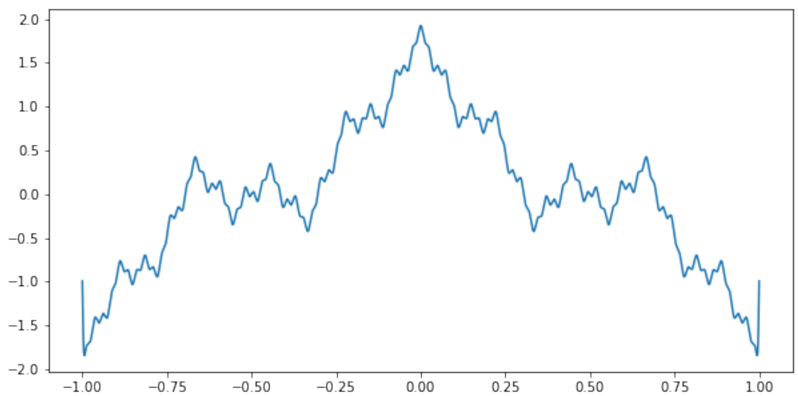

Figure 2: Filtered Weierstrass function.

References

[1] Wikipedia - Weierstrass Function