

Exercise 3: Sound processing

Multimedia systems

2018/2019

Create a folder `exercise3` that you will use during this exercise. Unpack the content of `exercise3.zip` that you can download from the course webpage to the folder. Save the solutions for the assignments as the *Matlab/Octave* scripts to `exercise3` folder. In order to complete the exercise you have to present these files to the teaching assistant. Some assignments contain questions that require sketching, writing or manual calculation. Write these answers down and bring them to the presentation as well. The tasks that are marked with ★ are optional. Without completing them you can get at most 75 points for the exercise (the total number of points is 100 and results in grade 10). Each optional task has the amount of additional points written next to it, sometimes there are more optional exercises and you do not have to complete all of them.

Introduction

In this exercise, you will generate simple sounds, vary their parameters and perform frequency analysis. You will also familiarize yourself with basic audio filtering and effects.

Assignment 1: Generating sounds

The first assignment will focus on generating simple waveforms, plotting them and playing them via speakers.

- (a) Generate a sine wave, display it and play it with the inbuilt function `sound`. The sine wave is a function of time

$$y = A \sin(\omega t + \varphi) \quad (1)$$

where A is the amplitude (from 0 to 1), ω is the angular frequency (i.e. the frequency in Hz multiplied by 2π), and φ is the phase (in radians). Use the standard sampling frequency of 44.1 kHz. That means that you have to calculate the value of the waveform 44100 times for each second of your recording.

Question: What do the formula parameters mean in analytical terms and how does that reflect in the sound produced by the formula?

- (b) Sounds encountered in real life situations are never as clean as the sinusoids you generated in the previous assignment. Try adding some noise to the waveform, then plot and listen to the result. Experiment with different types of noise.

- (c) Harmonics are what gives different instruments their sound color or timbre. They are softer multiples of the primary frequency. Try adding multiples of the primary frequency at a lower amplitude to your sinusoid and listen to it. Experiment with odd and even multiples.

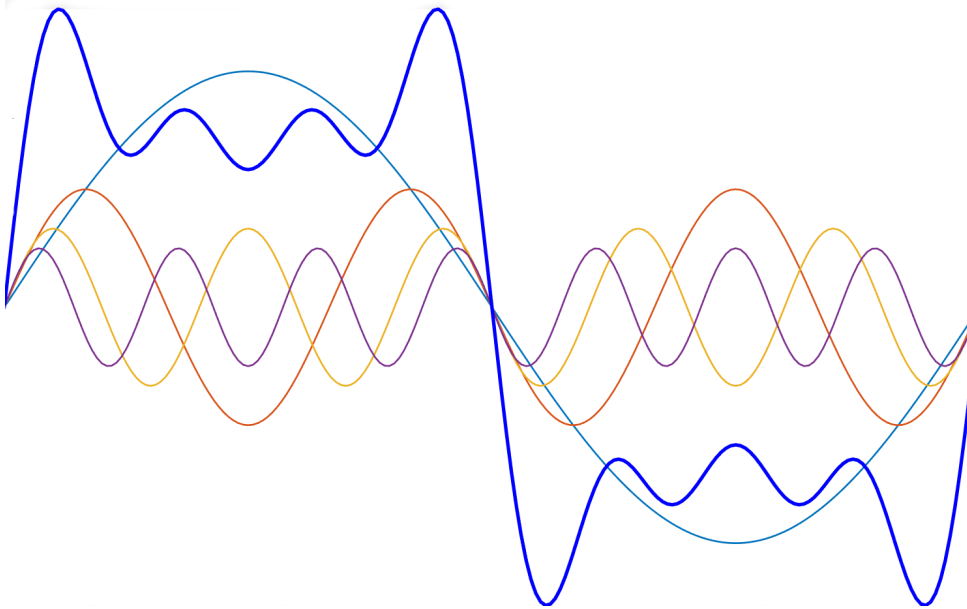


Figure 1: Harmonic frequencies and the final signal in blue.

- (d) ★ (10 points) Write a function to generate non-sinusoidal waveforms of your choice like square, triangle or sawtooth. You can also experiment with more exotic sounds, like chirp. Implement at least 3 different sounds.

Assignment 2: Frequency analysis

Due to the high sampling rates, visually interpreting the digital audio signal is usually difficult. Transforming the signal to the frequency spectrum allows us to interpret the signal content more directly.

- (a) Calculate the Fourier transform of a simple waveform using the inbuilt function `fft`. You also need to divide the result by the number of points used for the FFT, which is equal to the signal length by default. Since the result is complex and symmetrical, you will only use the positive part to plot the frequency components. Take the absolute value of the result and then either use the first $F_s/2$ values (where F_s is the sampling rate) or use the function `fftshift` to get useful values. The resulting spectrum should go from 0 to $F_s/2$, which is the highest theoretical frequency that can be contained in the signal (per the Nyquist theorem). Plot the results for all signals you generated in Assignment 1.

Help yourself with the documentation at <http://www.mathworks.com/help/matlab/ref/fft.html>

Question: How do the formula parameters influence the frequency spectrum?

- (b) ★ (5 points) Aliasing can occur when the signal is sampled too sparsely, which causes high frequencies included in the signal to reflect back to lower spectrum and produce errors in the frequency analysis. Use one of the signals from Assignment 1 and sample it with a frequency below the Nyquist frequency (i.e. the sampling rate should be lower than twice the highest frequency present in the signal). Calculate and plot the frequency spectrum.

Question: Considering the human hearing range, does the standard sampling frequency of 44.1 kHz seem arbitrary?

Assignment 3: Filtering

Audio signals can be processed to extract or attenuate certain frequency ranges. Since the design of audio filters is a large field, you will only focus on simple low- and high-pass Gaussian filters and their effects on audio signals.

- (a) Use the included function `gaussian_kernel` to calculate a kernel for performing a low-pass operation on an audio signal. Use the function `conv` to perform the filtering. Plot and listen to the result. Choose a signal that will produce obvious results.
- (b) Convert the low-pass Gaussian kernel into a high-pass filter. This can be achieved by alternately multiplying the kernel coefficients by `-1`. The resulting kernel will remove the low frequency components of the signal and only retain the high frequencies. Test on a sound of your choice. You can also use inbuilt sounds (`load gong`; `load handel`).

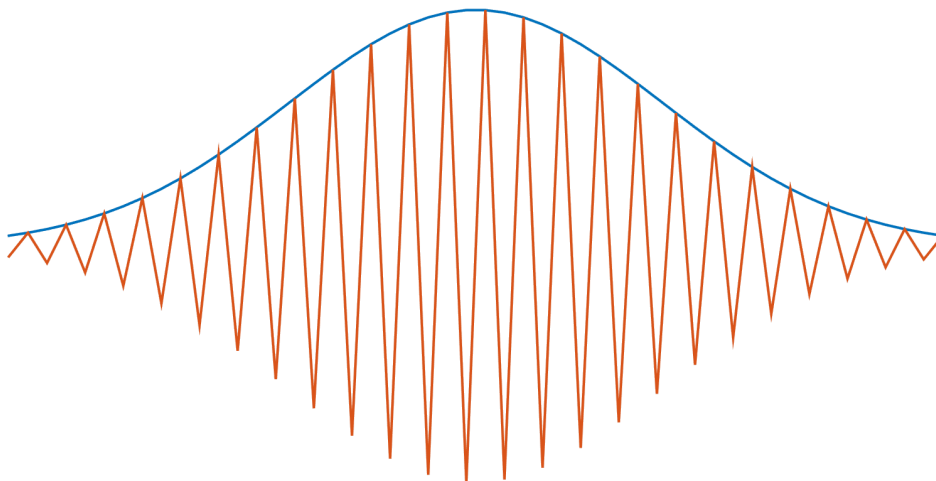


Figure 2: Low-pass and high-pass Gaussian filters

Assignment 4: Effects

Special kinds of filters can also produce other effects. Here you will implement some of them.

- (a) Delay: A delay time-shifts the signal and adds it to itself. Write a function that introduces a delay of a specified duration. You can again use the function `conv`.

- (b) Echo: Echo is a combination of multiple delays combined with attenuation. Write a function that accepts the number of echoes and their corresponding damping factors. Display and play the results.
- (c) ★ (10 points) Flanger: Is an effect produced by introducing a delay which depends on an outside oscillator. Put simpler, the delay for each sample of the output is not constant but changes based on a sinusoidal function.
- (d) ★ (10 points) Distortion: This effect changes the frequency content of the signal by adding gain to high energy frequencies and thus producing clipping. Implement it by using the formula from the lecture slides

$$y(n) = \frac{(1 + k)x(n)}{1 + k |x(n)|} \quad (2)$$

where k controls the amount of distortion.

Question: How do these effects change the signal in both time and frequency domains?