BRNO UNIVERSITY OF TECHNOLOGY FACULTY OF INFORMATION TECHNOLOGY

# Signals and Systems

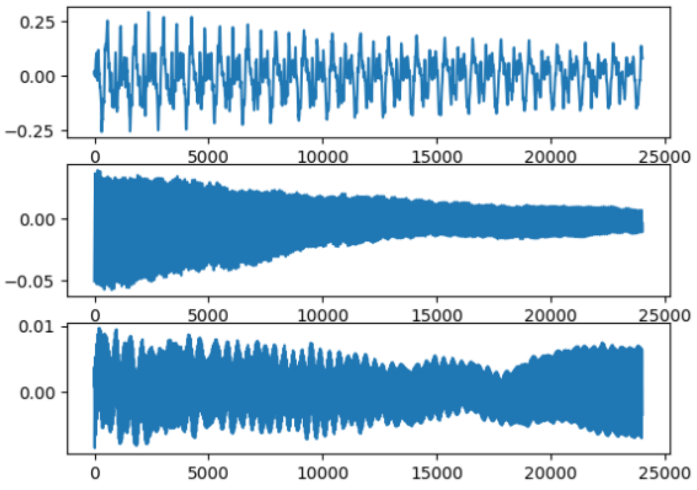
## Project 2022/23

Maksim Kalutski (xkalut00)

December 18, 2022

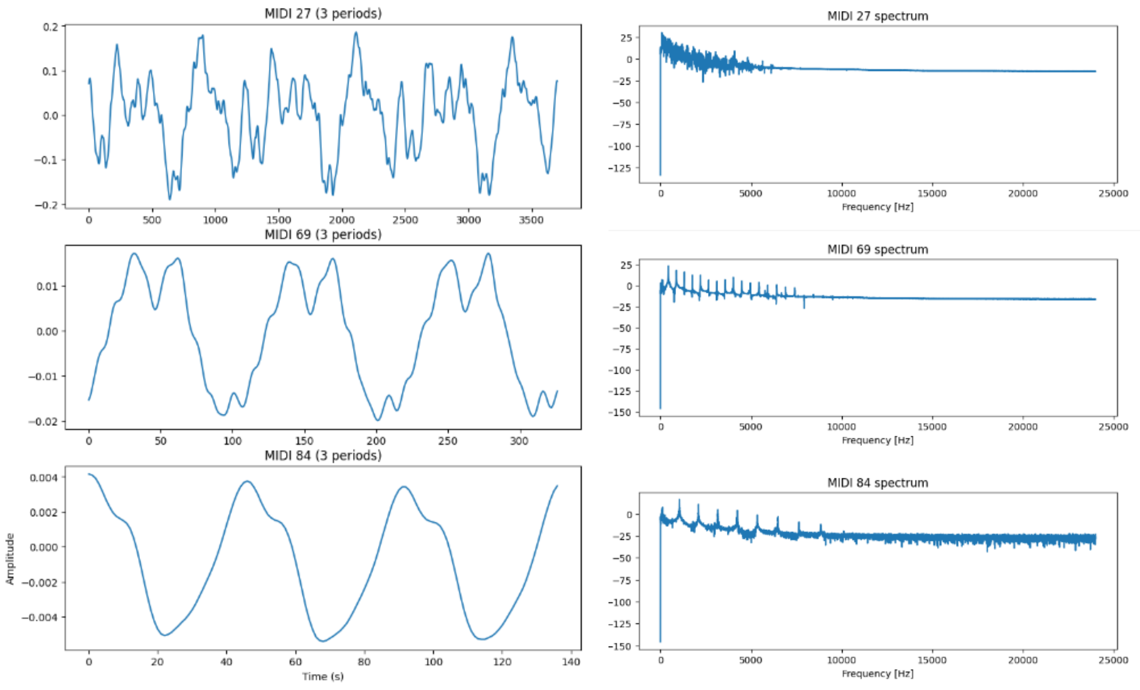
## 4.1 Basics

First of all, I'm showing the graph for each of my tones (27, 69 and 84) and saving them in the audio folder for later use.



Then I am plotting 3 periods for of my three tones. I loop through a list of *tones*. Each tuple contains a MIDI note number (27, 69, 84) and a frequency value (38.89, 440.00, 1046.50). For each loop iteration, the code calculates the number of tones skipped from a value called *MIDIFROM* and determines the start and end samples for a section of audio based on this value and the frequency of the current tone.

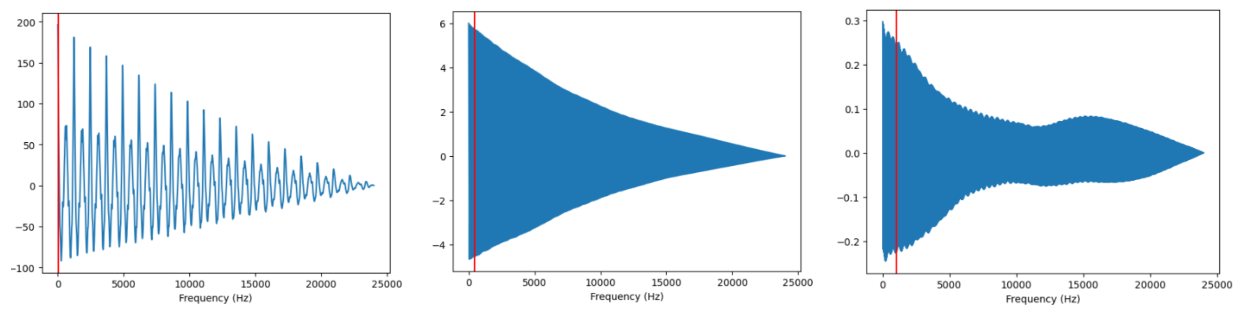
After that I plot the spectrum for each tone. I'm performing a fast Fourier transform on an input signal compute the magnitude and phase spectrum of the transformed signal. Maybe it would be better to move the signal a bit, but I haven't figured a way how to do it in Python.



## 4.2 Determination of the fundamental frequency

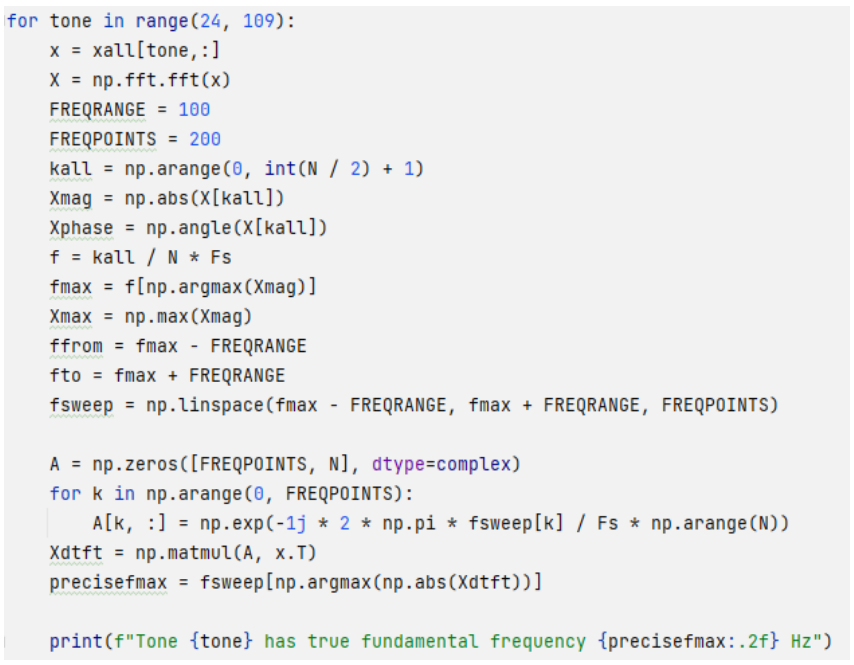
I use *‘scipy.signal’* module to calculate the autocorrelation of the audio data for each tone, and then find the peaks in the autocorrelation. The fundamental frequency can be calculated by dividing the sample rate (Fs) by the distance between the peaks. Parameter *distance* means that only peaks that are at this distance will be detected. The code then prints out the fundamental frequency for each tone. When comparing with the frequencies in midi.txt, we can notice that with each tone the calculation becomes more and more inaccurate.

Then I plot a graph for each of the tones with the method used and mark $f_o$ that I found using autocorrelation on it.



## 4.3 Refinement of the estimate of the fundamental frequency $f_o$

To implement DTFT I first transform signal into the frequency domain using the Fast Fourier Transform (FFT). Extract the magnitude and phase of the resulting frequency spectrum. Find the maximum frequency in the spectrum and determine a range around this frequency. Finally, I perform a Discrete Time Fourier Transform (DTFT) on the signal over a frequency range centered on the previously determined maximum frequency, and print the refined estimate of the fundamental frequency of the signal.



```
for tone in range(24, 109):
    x = xall[tone,:]
    X = np.fft.fft(x)
    FREQRANGE = 100
    FREQPOINTS = 200
    kall = np.arange(0, int(N / 2) + 1)
    Xmag = np.abs(X[kall])
    Xphase = np.angle(X[kall])
    f = kall / N * Fs
    fmax = f[np.argmax(Xmag)]
    Xmax = np.max(Xmag)
    ffrom = fmax - FREQRANGE
    fto = fmax + FREQRANGE
    fsweep = np.linspace(fmax - FREQRANGE, fmax + FREQRANGE, FREQPOINTS)

    A = np.zeros([FREQPOINTS, N], dtype=complex)
    for k in np.arange(0, FREQPOINTS):
        A[k, :] = np.exp(-1j * 2 * np.pi * fsweep[k] / Fs * np.arange(N))
    Xdtft = np.matmul(A, x.T)
    precisefmax = fsweep[np.argmax(np.abs(Xdtft))]

    print(f"Tone {tone} has true fundamental frequency {precisefmax:.2f} Hz")
```