# # I. Research process description

## ## Experiment Plan

1. Single-note waveform analysis and instrument accuracy level

2. Multi-note waveform analysis and instrument accuracy level

## ## Input data

The experiment inputs shall be from various instruments through non-professional input devices (microphones).

## ## Output validation

The experiment outputs shall be validated against instrument tuners.

## ## Algorithmic model

We assume that T is a vector containing the waveform of a recorded sound interval of length CHUNK with an input rate of RATE.

We apply Fast Fourier Transformation (FFT) on T, thus we get a vector of complex numbers C.

We calculate the absolute value of the elements of C, and we get a vector of real numbers P.

The indexes of P are the separate sound waves, and their values are the loudness of that wave.

We apply a noise gate to P. eliminating background noise under a set strength X (setting value to 0).

We define M as a vector of the indexes of peaks in P.

For M we calculate H = | M / CHUNK \* RATE |, which is a vector of the ringing frequencies in the sound chunk.

All in one:

H = abs(peaks(noise\_gate(X, abs(fft(T)))) / CHUNK \* RATE)

# # II. Case study

## ## Trial Experiment 1: Single-note identification

### ### Input

1. Baritone Ukulele (4 strings)

2. Generated Frequency (single frequency)

### ### Validation

Frequency table (example: https://pages.mtu.edu/~suits/notefreqs.html)

### ### Results

Experiment\_1.xlsx

Individual frequencies are identified with a +-6 Hz accuracy error.

The +-6 error is insignificant in higher pitch contexts, but may lead to incorrect identification on the lower end.

This is due to the distance between notes decreasing as they approach 0 (8 times every octave),

and the usual home microphone not being able to pick up bass notes.

### ### Code

live\_freq.py

### ### Observations

Due to instruments resonating single notes in higher octaves as well, we will consider

only the lowest identified frequency when searching for single notes.

Reducing the values after the FFT leads to better nose cancellation

(in general / RATE \* CHUNK does the trick instead of a constant).

Noise is magnitudes stronger in the lower end of the frequency spectrum.

# # Related work

## ## Bearing vibration detection and analysis using enhanced fast Fourier transform algorithm

### ### Authors

Hsiung-Cheng Lin, Yu-Chen Ye, Bo-Jyun Huang - October 2016

### ### Link

https://journals.sagepub.com/doi/full/10.1177/1687814016675080

### ### Comparison

The paper describes the use of the FFT algorithm and an enhanced e-FFT algorithm in the use of

identifying faults in mechanical bearings.

Due to the nature of extremely quiet vibrations given off by the bearing, they were in need of e-FFT for noise cancellation, and have identified frequencies with high accuracy.

In our work, frequencies are well defined and powerful, thus we are not in need of e-FFT, and normal FFT does the job.

## ## Audio Analysis using the Discrete Wavelet Transform

### ### Authors

George Tzanetakis, Georg Essl, Perry Cook

### ### Link

https://soundlab.cs.princeton.edu/publications/2001\_amta\_aadwt.pdf

### ### Comparison

The paper tackles quite a similar problem to ours, but using a different base algorithm for identifying notes.

The procedure involves pattern recognition and low level AI, that we would like to omit and focus on a pure mathematical approach.

# GITHUB

started working 4 days before the deadline: <https://github.com/lippaybalazs/ResearchProject>

