TTT4295 - Acoustic Signal Processing

Assigment 1

A Music Box As A Sound Source

performed by

Njaal Svensen
Kristian Goeystdal
Lina Andersson
Josh Jude

Report by

Josh Jude

Department of Electronic Systems
Norwegian University of Science and Technology

Summary

This report investigates the spectral properties of a mechanical music box using FFT-based analysis. Audio was recorded at a sampling rate of $f_s = 44.1$ kHz and segmented into 53 notes, from which one of these note clips was further analyzed. An FFT of size $N = 65536$ resulted in a frequency resolution of $\Delta f = 0.673$ Hz. Estimation of the fundamental frequency f_0 and classification of peaks was done by splitting the peaks into two groups, harmonic (group A) or non-harmonic (group B).

The measured fundamental frequency was 699.829 Hz, which was +3.6 cents relative to the theoretical 698.460 Hz. This corresponds to the note F_5. The second and third harmonics lay within ± 0.675 Hz of the ideal multiples and were measured at -3.3 dB and -5.5 dB relative to the fundamental frequency. The fundamental frequency followed the equal-tempered scale accurately, and the relative harmonic frequency levels decayed with the order as expected. Non-harmonic components were observed and are attributed to the mechanical resonances and minor tuning errors of the box itself.

Contents

1 Introduction **1**

2 Theory **2**

3 Method **3**

4 Results **4**

5 Discussions **6**

6 Conclusions **7**

A Appendix **9**

A.1 Code for frequency analysis 9

1 Introduction

A music box is a small instrument that generates musical notes by using a small set of metal pins. It consists of a rotating drum, and a comb with metal prongs. The drum is rotated, and the pegs on the drum push and release different combinations of the metal prongs. This movement causes strong vibrations in the prongs, such as in cantilever beams. Each prong generates its own fundamental frequency and overtones, which is the source of the audible notes the user can hear.

2 Theory

To analyze the spectral content of the audio coming from the music box, a very useful approach is to process the data using a FFT, or *Fast Fourier Transform.* The FFT is an efficient way to compute the DFT, or *discrete Fourier transform* of a signal. A DFT converts time-domain signals into their equivalents in the frequency domain, revealing the frequencies present in the signal [1].

There could be inaccuracies in the measurement process given an FFT of size N, known as the frequency uncertainty. Given a sampling rate f_s, the spacing between frequency bins corresponds to this uncertainty, and is described in eq.(2.1).

$$\Delta f = \frac{f_s}{N} \tag{2.1}$$

The well-tempered musical scale was used to map the frequencies to notes. Given the reference pitch of the A_4 semitone, 440 Hz, the fundamental frequency for a semitone n within this octave can be found using eq.(2.2).

$$f_n = 440 \cdot 2^{\frac{n-9}{12}} \text{ Hz} \tag{2.2}$$

Table 2.1 shows the 12 semitones for octave no. 4 [2]. To find the fundamental frequencies for k octaves above or below, the fundamental frequency in the table can be multiplied by 2^k where k is positive for higher octaves and negative for lower octaves.

Table 2.1: The fundamental frequencies of the 12 semitones in the 4th octave.

n	**Note**	$\mathbf{f_{fundamental,n}}$ **[Hz]**
0	C_4	261.63
1	$C\sharp_4$	277.18
2	D_4	293.66
3	$D\sharp_4$	311.13
4	E_4	329.63
5	F_4	349.23
6	$F\sharp_4$	369.99
7	G_4	392.00
8	$G\sharp_4$	415.30
9	A_4	440.00
10	$A\sharp_4$	466.16
11	B_4	493.88

When measuring the frequencies of the notes from the music box, there might be inconsistencies between the ideal fundamental frequency, f_{ref} and what is measured, f_0. This difference can be measured in cents, c, using eq.(2.3) [2].

$$c = 1200 \cdot \log_2 \frac{f_0}{f_{ref}} \tag{2.3}$$

The relative harmonic levels are expressed in decibels using Equation (2.4) [2], where mag_h represents each harmonic's amplitude and mag_f represents the fundamental frequency amplitude,

$$l = 20 \log_{10} \frac{\text{mag}_h}{\text{mag}_f} \tag{2.4}$$

3 Method

To collect the audio, a smartphone with a sampling rate of $f_s = 44.1$ kHz was used to record the music box. The music box was cranked slowly to allow each note to die out completely before the next one started. After recording, the audio clip was then manually split into 53 different sections, and further processing was done on each individual clip.

An FFT size of 65536 was used for every recording. Frequency analysis was conducted using numpy's FFT functions. To find the fundamental frequency, the peak with the highest magnitude in the frequency spectrum was found and denoted f_0. The other peaks in the spectrum were sorted into harmonic (group A) or non-harmonic (group B) categories based on their relationship to f_0.

The equations (2.3) and (2.4) were implemented in the code to show both the relative levels of the harmonically related peaks and the deviations from a perfect harmonic relationship.

4 Results

The figures 4.1 and 4.2 show the waveform and the frequency spectrum from an analysis done of a single note recorded from the clip, respectively.

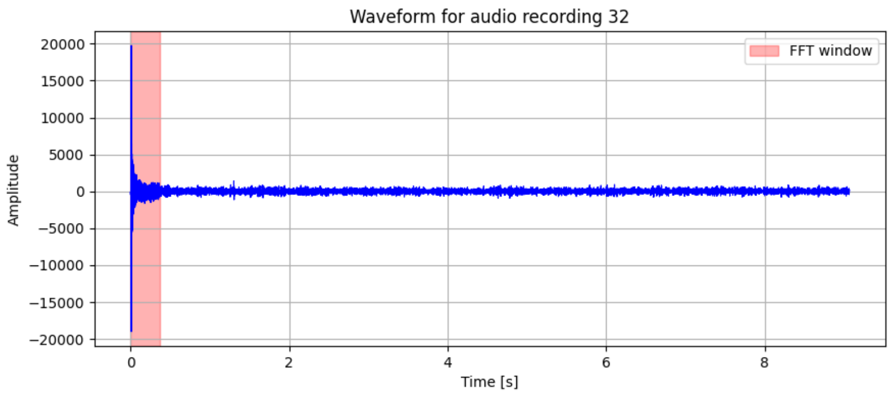

Figure 4.1: The waveform of recording no. 32, along with a marked section showing what part was used for the FFT.

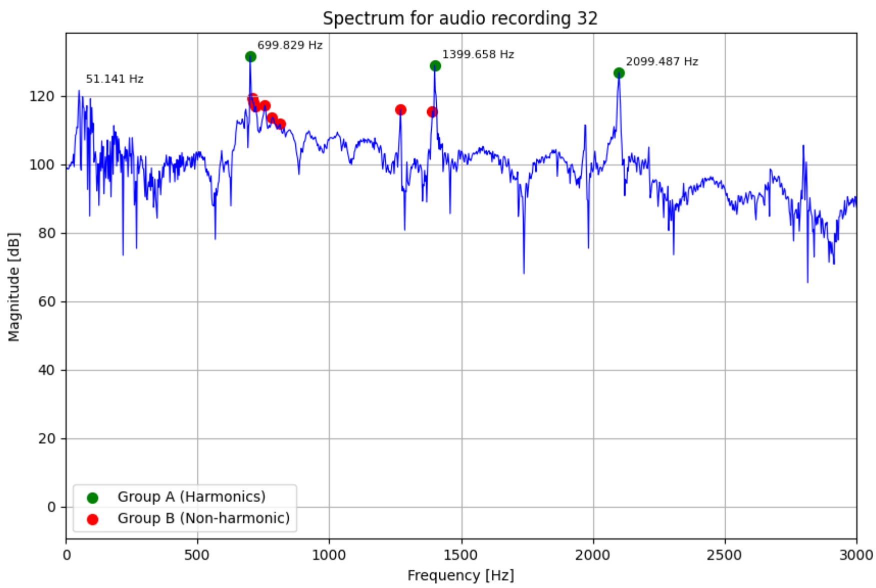

Figure 4.2: The frequency spectrum of recording no. 32, showing which peaks belong in group A or B.

As mentioned in section 3, the FFT size N was set to 65536 with the sampling frequency f_s

set to 44.1 kHz. Using equation (2.1), the frequency uncertainty Δf was calculated to be equal to 0.673 Hz.

Table 4.1 shows the relationships between the harmonic frequencies (group A) found in the audio clip, with the deviations both in Hz and in cents according to equation 2.3, and the relative levels in decibels.

Table 4.1: Frequency analysis of recorded note no. 32.

#	**Frequency [Hz]**	**Dev [Hz]**	**Dev [cents]**	**Level [dB]**
Group A (Harmonics): 3 peaks				
f_0	699.829	+0.000	+0.00	+0.00
$2f_0$	1400.331	+0.675	+0.85	-3.30
$3f_0$	2098.814	-0.675	-0.55	-5.48

By looking at Table 2.1, the fundamental frequency of the note F_4 can be mulitplied by 2, which results in a fundamental frequency of 698.460 Hz. This corresponds to the note F_5 with a deviation from the fundamental frequency found in the signal by 1.37 Hz. Equation (2.3) can be used to find the differences in cents between the theoretical fundamental frequencies and those present in the signal. Table 4.2 shows those differences.

Table 4.2: Deviation in cents between theoretical and measured frequencies.

Harmonic	**Theoretical [Hz]**	**Measured [Hz]**	**Deviation [cents]**
f_0	698.46	699.829	+3.60
$2f_0$	1396.92	1400.331	+4.24
$3f_0$	2095.38	2098.814	+2.84

5 Discussions

The measured fundamental frequency in the signal was off by the theoretical frequency of the note F_5 by 1.368 Hz. The deviation was $+3.6$ cents, which can be considered a negligble difference for most humans [3]. The harmonic frequencies found are as expected nearly exact integer multiples of the first fundamental freqency, with offsets equal to the frequency uncertainty of 0.675 Hz. This indicates that the offset is a result of the signal itself, and not any property of the measurement process.

The relative levels in decibels show a consistent decay of the harmonic frequencies, with the higher-order multiples showing a greater attenuation. As seen in Figure 4.2, there are many non-harmonic peaks present in the signal. This is most likely due to the mechanical artifacts of the music box, or slight tuning errors.

6 Conclusions

The analysis of the audio shows that the music box produces tones that match the theoretical musical scale nearly perfectly. The frequency deviations that were present were within a range that is not audibly differentiable for untrained ears. The harmonics within the frequency spectrum followed the expected pattern, with a strong fundamental frequency and weaker integer multiples of that frequency present as well. The music box contains mechanical imperfections that introduce additional non-harmonic components, but these do not affect the notes in any significant way. The music box can therefore be an accurate but simple acoustic source of notes.

Bibliography

[1] Svensson, Peter, TTT4295 - Acoustic Signal Processing, 16 Sep 2024, Department of Electronic Systems, Trondheim, 2024, 160 p.

[2] Svensson, Peter, Assignment 1 - Analyze the music box as a sound source, 2 Sep 2025, Department of Electronic Systems, Trondheim, 2025, 5 p.

[3] Loeffler, Beatus Dominik, Instrument Timbres and Pitch Estimation in Polyphonic Music, May 2006, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, 2006, 91 p. Available at: `https://web.archive.org/web/20071218232401/http://etd.gatech.edu/theses/available/etd-04102006-142310/`

A Appendix

A.1 Code for frequency analysis

```
## Code for Assignment 1 in TTT4295, autumn 2025
## Josh Jude

import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import find_peaks
from scipy.io import wavfile
import os
import glob

def analyze_harmonics(filename, threshold=0.1, n_fft=65536, fmin=20.0, plot=False):
    rate, data = wavfile.read(filename)
    n = len(data)

    if n < n_fft:
        x = np.pad(data, (0, n_fft - n))
    else:
        x = data[:n_fft]

    X = np.fft.fft(x)
    freqs = np.fft.fftfreq(n_fft, 1/rate)
    mag = np.abs(X)
    freqs = freqs[:n_fft // 2]
    mag = mag[:n_fft // 2]

    peaks, _ = find_peaks(mag) # finding the local maxima
    max_mag = mag[peaks].max() # strongest peak magnitude
    strong = [p for p in peaks if mag[p] >= threshold * max_mag] # keeping peaks
        above a certain threshold
    strong_sorted = sorted(strong, key=lambda i: freqs[i]) # sorting by frequency
    strong_sorted = [i for i in strong_sorted if freqs[i] >= fmin] # discarding
        weaker peaks

    max_peak_idx = max(strong_sorted, key=lambda i: mag[i])
    f0 = freqs[max_peak_idx] # fundamental frequency is the largest magnitude peak
    mag0 = mag[max_peak_idx] # magnitude of fundamental peak

    tolerance_hz = rate/n_fft # the bin width

    groupA, groupB = [], [] # splitting into harmonics and non-harmonics
    for idx in strong_sorted: # checking every frequency that was kept
        f = freqs[idx] # peak freq
        if f < f0 - tolerance_hz: # skipping if the freq is below fundamental
            continue
        k = int(round(f / f0)) # finding the harmonic number
        if k < 1: # adding to group B if less than fundamental
            groupB.append({"frequency": f, "magnitude": mag[idx]})
            continue

        f_ideal = k * f0 # calculating the deviation from an ideal harmonic
            frequency
        dev_hz = f - f_ideal
        if abs(dev_hz) <= tolerance_hz: # if it is close enough, calculate the
            deviations and add to group A
```

```
            dev_cents = 1200.0 * np.log2(f/f_ideal)
            level_db = 20.0 * np.log10(mag[idx]/mag0)
            groupA.append({
                "k": k,
                "frequency": f,
                "magnitude": mag[idx],
                "deviation_hz": dev_hz,
                "deviation_cents": dev_cents,
                "level_db_rel_f0": level_db
            })
        else:
            groupB.append({"frequency": f, "magnitude": mag[idx]}) # if not close enough, add to group B

    return {
        "f0": f0,
        "tolerance_hz": tolerance_hz,
        "groupA": sorted(groupA, key=lambda d: d["k"]),
        "groupB": groupB
    }
```
Listing A.1: Code for Assignment 1 in TTT4295, autumn 2025