Experimental Methods and Signal Processing (CEE 5984)

FINAL PROJECT PROPOSAL – MUSICAL INSTRUMENT TUNING APP

V0.1: 2018-11-10

Martin Scavone

Background and project problem statement

Chord musical instruments rely on a set of tense strings to produce sound. When put to vibrate by the player, these will emit sound at certain predefined frequencies – those of the musical notes. However, in a similar manner to any material subject to constant tension over a long period of time, the instrument's strings will tend to creep, releasing themselves of such tensile stress (stretching themselves) and so they would lose their tuning to the note they were meant to vibrate at (Lynch-Aird and Woodhouse, 2017). That is why such instruments must be periodically checked and tuned before playing, and even the reason why automatic tuning devices exist.

The task of tuning consists of applying or releasing tension to the instrument's strings so that, when operated by the player, they will produce sound at the frequency of certain musical notes. A variety of professional-level tuning aids are available for musicians – even smartphone applications that use the phone's built-in microphone to listen to a musical instrument and evaluate whether the instrument is tuned or not[1].

Voice range for human singers, however, cannot be tuned, but the classification of voices is similarly frequency-based[2] - a lyric singer must be able to cover a predefined frequency range with their voice to be entitled to any voice range. Nevertheless, both tuning a chord instrument and recognizing a singer's voice range involve analyzing the frequencies at which both subjects' sounds are emitted – a task that can be performed with the aid of a computer.

On that grounds, this document will present the work plan towards the implementation of a Matlab-based musical instrument and voice range recognition app, to be run on any personal computer. Besides, an ancillary experiment will be conducted to test how microphones with different frequency ranges would perform when put to listen to sounds on the fringe and out of such ranges.

Project proposal.

This project will consist of two main experiments – namely *Experiment 01,* the development of the musical instrument tuner app, and *Experiment 02*, the comparison of the signals produced by different microphones after a sound on the verge or off their declared frequency range. This chapter is divided into three subsections: firstly, the statements that will be assumed true throughout the development of the project will be pointed out. This 'assumptions' section will be followed by the two experiments' objective statement, and finally, discussion will be brought on the applicability of these experiments' outcome.

Assumptions

The development of the musical instrument tuner will rely on the following axioms. These have been written referring to musical instruments, but they also will be assumed true for the human voice:

[1] An example of such an app is "Guitar Tuna", both available for Android, Windows, and iOS mobile phones

[2] Refer to the following archived music-school course entry for definitions on the most widespread male and female voice categories: https://web.archive.org/web/20080605192215/http://pioneer2.aaps.k12.mi.us:80/choir/voiceclass.html

1) The sound signals produced by both a subject instrument[3] can be processed by frequency analysis and will have peaks at certain frequencies, namely its fundamental frequency and its harmonics.

2) The fundamental frequency of the sensed sound (i.e. lowest frequency value at which an outstanding peak in magnitude is observed) is that of its matching musical note; whereas all the harmonics will define the instrument's timbre.

3) The sounds of the subject instruments will not contain sub-harmonics. In other words, no outstanding peaks will be expected at frequencies lower than the fundamental frequency. This assumption is based on what stated in the work by Berg and Stork (2005), where it is implied that very few instruments actually have sub-harmonics (the only example of such an instrument provided therein is a percussion instrument, and these will be disregarded in the proposed implementation for Experiment 01)

4) All the instruments will be tuned to the equally-tempered scale with the A_4 set at 440.00Hz. The equally-tempered scale features a fixed value to increase the frequency of any given note and its successor. In other words, the ratio of the frequencies of the two notes that define a semitone is constant – an equal to $\sqrt[12]{2}$. This scale has been selected since it is the most widespread tuning scale (Hartmann, 2013) However, that is not the case of pianos, which are purposefully tuned with an offset from the equally-tempered scale – each note's offset from the scale is given by the Railsback curve (Fletcher and Rossling, 1991). The inclusion of the Railsback curve in the tuner's programming can nevertheless be achieved with ease.

Experiment 01 – Development of a Matlab-based musical instrument tuning app

The goal of this Experiment 01 is to implement a Matlab-based application for tuning string instruments. Tuning will be based on real-time frequency analysis of incoming sounds; the player must play the instrument in the manner needed for tuning while the app is running so that the instrument's sounds could be recorded and processed. The app will report the player whether the note he or she is playing with the subject instrument matches a tuning note or not.

The proposed tuning application will also be able to compute the frequency range of any person's voice, by sampling their lowest and highest singing frequencies. Further details on how the frequency analysis of the input signals will be conducted are provided in the *Materials and Methods* chapter.

The instrument tuning application will be programmed in Matlab yet full compatibility with GNU-Octave will be sought.

Experiment 02 – Assessment of different microphones for musical instrument tuning purposes.

Not all microphones for PCs operate in the entire range of human hearing (20-20,000 Hz). In fact, there are commercially available microphones which operate in the 100-16,000Hz frequency band – designed for voice recording only and intended for Voice-over-IP [VoIP] applications. An end-user may wish to try the final product of Experiment 01 using a non-full-range microphone to tune any instrument, yet no accurate tuning could be guaranteed when the attempt is made of tuning notes beyond the microphone's frequency range.

[3]The term "subject instrument" will refer to the musical instrument that whose sounds are to be analyzed by the tuner

On such grounds, experiment 02 will consist of recording the sounds produced by a properly-tuned bass guitar when playing its tuning notes with full-range and wideband microphones and comparing the signals retrieved in terms of frequency content and attenuation.

Applicability

For the sake of this Project, the instrument tuning app will be developed to assist in the tuning of guitars, violins, bass guitars [4-string and 5-string], and pianos[4]. However, its analysis capabilities can easily be extended to other instruments -provided the notes these are tuned against are known – or even human voices: by performing frequency analysis on a person's voice, the application may solve the singer's vocal range.

Materials and Methods

Microphones

As it has been mentioned in previous paragraphs, Experiment 02 will consist of comparing how responsive different microphones are to the tuning notes of certain instruments – most particularly the lower tuning notes in guitars and bass guitars, whose frequencies are below 100Hz. The following microphones will be used to record the sounds of a tuned bass guitar while playing its tuning notes:

1. Laptop's built-in omnidirectional microphone array. (Specifications not available)
2. Typical wide-band plug-in microphone for VoIP applications. Frequency range 100-16000Hz. Sensitivity 60dB.
3. Full-bandwidth plug-in PC microphone. Frequency range 20-20000 Hz. Sensitivity 58dB.
4. Professional-level condenser microphone. Frequency range 20-20000Hz. Sensitivity 37dB

Aside from that, microphones 1 and 3 from the above list will be used throughout the beta-testing of the musical instrument tuning app (Experiment 01).

Computer software

Experiment 02 is not by any means overly demanding in terms of computing requirements – Matlab's built-in audio recording and frequency analysis functions are the most sophisticated software requirements for this experiment.

That, nonetheless, is not the case for Experiment 01, in which the programming of the instrument tuner -which will also rely on Matlab's built-in commands to operate- is the core challenge. That is the reason why this chapter will mostly deal with Experiment 01's software requirements. For the sake of simplicity, however, the sampling time, length and frequency resolution constraints that may be set for Experiment 01 will also be kept for Experiment 02.

Frequency range, frequency resolution, and time of sampling

Each of the compatible instruments has predefined tuning frequencies (see Table 1 for the case of guitars, 4 and 5-string bass guitars, and violins, and Table 2 for the frequency ranges defining each male and female voice category). Such would be the frequency range of each tuning process. In other words, that

[4]One of the project's key assumptions must be highlighted once again here: the musical instrument tuning app will assume that all instruments are tuned to an equally-tempered scale, whereas in fact the lower and higher octaves in pianos are "stretched" beyond the target frequency values of the equally-tempered scale (as pointed out for instance in Fletcher and Rossling (1991)).

would be the frequency range to be adopted each time the instrument tuner is called to tune one of such instruments.

Table 1: Frequencies of tuning notes of selected musical instruments

	ID	Freq [Hz]	Note Id
Guitar	1	82.41	E2
	2	110	A2
	3	146.83	D3
	4	196	G3
	5	246.94	B3
	6	329.63	E4
Violin	1	196	G3
	2	293.66	D4
	3	440	A4
	4	659.26	E5
4-string Bass guitar	1	41.204	E1
	2	55	A1
	3	73.416	E2
	4	97.999	G2
5-string Bass guitar	1	30.868	B0
	2	41.204	E1
	3	55	A1
	4	73.416	E2
	5	97.999	G2

Table 2: Frequency limits for male and female voice ranges

	Freq. Low	Freq. High	Note low	Note high	Range
Male adult voices	82.41	329.63	E2	E4	Bass
	110	440	A2	A4	Baritone
	130.81	523.25	C3	C5	Tenor
	164.81	659.25	E3	E5	Countertenor
Female adult voices	174.61	698.46	F3	F5	contralto
	220	880	A3	A5	Mezzo-soprano
	261.63	1046.5	C4	C6	soprano

All in all, the most demanding case scenario is the tuning of a piano, in which sounds with a frequency ranging between 27.50Hz and 4186.01Hz (disregarding the additional offset plotted in the Railsback curve) will be produced. This maximum expected frequency constrains the tuner's sampling frequency (f_s) must at the very least be twice larger than 4186Hz. So, a f_s of **16 kHz** will be used on every execution of Experiment 01.

Concerns about frequency range notwithstanding, the most crucial parameter to proper recognition of musical notes is the tuner's frequency resolution. In an attempt to resemble the most an experienced musician tuning a musical instrument, Experiment 01's tuner will aim at a frequency resolution (δf) of 1/230 of an octave – the frequency resolution of the human ear (Rebillard and Pujol, 2016). In terms of semitones (interval between two consecutive notes), **δf equals 5 cents** (5/100 of the difference in frequency of two consecutive notes). In the equally-tempered scale, the "size" of a semitone depends on the frequency of the two defining notes, and the smallest δf would be required when tuning the deepest note of pianos and five-string bass guitars (27.5 and 30.87 Hz respectively; leading to a **target δf of 1.72 and 1.53 Hz**).

A frequency resolution target in signal processing is only achieved by manipulating the length of the input signal in the time domain (Figliola and Beasley, 2015). That being said, the target δf would require a **sample time of 0.58 / 0.65 seconds**, in which the instrument's sound must be clearly noticeable. The implementation of Experiment 01 tuner will take samples at **0.7-second** intervals.

Fundamental frequency recognition and feedback to end-user

Experiment 01's tuner will conduct a fast Fourier transform of samples of the digitalized sound signals covering segments 0.7 seconds long. The note being played by the instrument (or in the case of voice recognition, the note at which the subject is singing) will be identified by the software as the smallest frequency value which produces a peak in the signal amplitude versus frequency plot. The tuner's peak recognition software will distinguish actual peaks in the frequency domain plot from peaks due to background noise or lesser frequency components by relying on the signal's mean and standard deviation, and the amplitude difference between the potential peak and the signal's mean amplitude.

Once the played note is identified by the tuning software, it will print this result on screen, along with the closest target tuning note and an indication of whether to tighten or release tension on the instrument's strings to achieve the target note.

References

- Berg, R. E., Stork, D. G. (2005): "The Physics of Sound". Third Edition. Pearson. ISBN: 013-145789-6
- Hartmann, W. M. (2013): "Principles of Musical Acoustics". Springer. ISBN: 978-1-4614-6786-1
- Fletcher, N. H., Rossling, T. D. (1991): "The Physics of Musical Instruments". Springer-Verlag. ISBN: 978-0-387 -94151-6
- Figliola, R. S., Beasley, D. E (2015): "Theory and Design for Mechanical Measurements". Sixth Edition. Wiley. ISBN 978-1-118-88127-9
- Rebillard, G.; Pujol, R. (2016): "The Transfer OF Airborne Sound Vibration to the Structures and Fluid of the Cochlea". Web article, available at http://www.cochlea.eu/en/cochlea/function. Last accessed 2018-11-07.
- Lynch-Aird, N.; Woodhouse, J (2017): "Mechanical Properties of Nylon Harp Strings". Materials, 2017, 10, 497. Pp. 1-32