On the Requirement of Automatic Tuning Frequency Estimation

Alexander Lerch
zplane.development
Berlin, Germany
lerch@zplane.de

Abstract

The deviation of the tuning frequency from the standard tuning frequency $440\,Hz$ is evaluated for a database of classical music. It is discussed if and under what circumstances such a deviation may affect the robustness of pitch-based systems for musical content analysis.

Keywords: concert pitch, tuning frequency, detuning.

1. Introduction

Pitch extraction from musical audio signals is an important task in the field of musical content analysis of monophonic as well as polyphonic input data. It is a required processing step for automatic transcription, melody finding, harmony and key detection and other algorithms for audio content analysis.

For these applications, various approaches to fundamental frequency detection have been proposed, but the mapping of frequencies to pitches is frequently regarded to be trivial, assuming the mid frequencies of the pitches to be tuned with reference to a standardized tuning frequency of $440\,Hz$ for the pitch A4.

On the other hand, there exist a few publications that address the issue of possible deviations of the real tuning frequency from $440\,Hz$ and propose algorithms for the automatic detection of this tuning frequency (see section 4.1).

This raises the question if an automatic tuning frequency detection could possibly improve the pitch tracking results or if its influence is negligible. To verify the hypothesis that it might improve the results, a pre-test with a simple automatic key detection has been executed on a small database (65 tracks) of key labeled jazz recordings. The correct classification rate increased from 70.8% at a fixed tuning frequency of $440\,Hz$ to 76.9% with adaptive tuning frequency estimation as described below. Although this result is statistically not significant due to the small test database, it indicates that pitch-based analysis systems may benefit from an automatic detection of tuning frequency.

2. Tuning Frequency

The concert pitch or standard (musical) pitch is used for tuning one or more musical instruments and is defined to be the pitch A4. Its frequency, the tuning frequency, is standardized internationally to $440\,Hz$ [1], but the exact frequency used by musicians can vary due to various reasons, e.g. the usage of historic instruments or timbre preferences, etc.

Over time, the variation of the tuning frequency decreased. Briner [2] mentions some typical frequency ranges for the tuning frequency from the past three centuries, displayed in Table 1 as deviation from $440\,Hz$:

Table 1. Typical Deviation of the tuning frequency from $440\,Hz$ over three centuries.

Year	lower deviation	upper deviation
~ 1750	$-50\,Hz$	$+30\,Hz$
~ 1850	$-20\,Hz$	$+20\,Hz$
~ 1950	$-5\,Hz$	$+10\,Hz$

Nowadays, while for the majority of electronic music productions the default tuning frequency of $440\,Hz$ can be assumed, the tuning frequencies of orchestras may show deviations from $440\,Hz$. For example, the Chicago Symphony Orchestra and the New York Philharmonic tune at $442\,Hz$, while the Berlin and Vienna Philharmonic orchestras have a tuning frequency of $443\,Hz$[1]. At least in the case of both European orchestras, the tuning frequency was higher in previous decades. The frequencies $442\,Hz$ and $443\,Hz$ correspond to deviations from the standard tuning frequency of $7.85\,cent$ and $11.76\,cent$, respectively. Such deviations may also occur in other music styles, especially when acoustic instruments are used.

Besides this intended shift of the tuning frequency, there may be unintended variations over the time of a concert or a recording session. For example, the tuning frequency could be slowly decreasing, as it can be sometimes recognized with choirs without accompaniment; contrarily, a rising involvement of the musicians during the concert could lead to an increasing tuning frequency. In the case of professional musicians, the maximum variation can probably be assumed to be three to five $cent$.

[1] personal communication with the orchestra's archivists, March and April 2006

3. Other Frequency Deviations

In the context of pitch analysis, there are also other possible reasons for detection inaccuracies that may add together with deviations of the tuning frequency. Partly, these are under control of the developer like the system's frequency detection accuracy. On other deviations, the developer has no or only limited influence.

3.1. Deviation of harmonics from equal tempered scale

In several applications, e.g. when creating a simple pitch chroma [3], the "pitch detection" is not only based on the detected fundamental frequency, but on all spectral maxima. In this case, the deviation of harmonics from the equal tempered scale (the scale they are mapped to) has to be taken into account. Table 2 shows mean and maximum deviation of the harmonic series from the closest equal tempered pitch frequency with respect to the number of harmonics. The fundamental is in tune with the scale.

Table 2. Deviation of harmonics from equal tempered scale

#Harm	max. deviation	mean abs deviation
3	$2.0\,cent$	$0.7\,cent$
5	$14.7\,cent$	$3.1\,cent$
7	$31.2\,cent$	$7.0\,cent$

While a maximum deviation of $31.2\,cent$ sounds alarming, its influence should not be overrated since the seventh harmonic usually has a small level compared to the level of lower harmonics, dependent on the instrument playing.

Furthermore, the deviation does not matter in many cases at all, since many systems for frequency tracking take the harmonic structure into account.

3.2. Deviation due to non-equal temperament

In the equal tempered scale, the frequency ratios of all intervals are multiples of $\sqrt[12]{2}$. In an analysis context, equal temperament is usually assumed, which is the only possible assumption since the key of the piece is in most cases unknown. However, a musician not restricted to the equal tempered scale by his or her instrument or accompaniment will most likely perform on a non-equal tempered scale, since the equal tempered scale is only a mathematical construct to make interval ratios independent from position and key. For example, two string instruments playing a fifth will most likely play a perfect fifth rather than an equal tempered one, just because it sounds more "natural".

The Pythagorean temperament (*PT*) and the Meantone temperament (*MT*) are used as examples to illustrate deviations from the equal tempered scale. Basically, *PT* is constructed with perfect fifths, while *MT* is constructed with perfect thirds. Table 3 shows the maximum deviations of the *PT* and *MT* scale from the equal tempered scale in cent with an A4 tuning frequency of $440\,Hz$ for different keys.

Table 3. Max. deviation of *PT* and *MT* from equal tempered scale

Key	max. deviation (*PT*)	max. deviation (*MT*)
C	$9.8\,cent$	$17.1\,cent$
D	$5.9\,cent$	$10.3\,cent$
E	$9.8\,cent$	$17.1\,cent$
F	$11.7\,cent$	$20.5\,cent$
G	$7.8\,cent$	$13.7\,cent$
A	$7.8\,cent$	$13.7\,cent$
B	$11.7\,cent$	$20.5\,cent$

4. Evaluation of Real World Signals

The mentioned deviations are within an assumed tolerance range of $\pm 50\,cent$, but they can add together. To be able to draw conclusions if an algorithms performance may be influenced by an incorrect tuning frequency assumption, the amount of tuning frequency deviation in real world signals has to be investigated. To get results for a large amount of test files, this analysis has to be done in an automated way.

4.1. Automatic Tuning Frequency Detection

The following systems have been proposed to find the best tuning frequency match automatically:

Scheirer [4] used a set of narrow bandpass filters with their mid frequencies at particular bands that have been handpicked to match pitches from the analyzed score. These filters are swept over a small frequency range. The estimated tuning frequency is then determined by the frequency of the maximum filter output sum.

Dixon [5] proposed to use a peak detection algorithm in the FFT domain, calculating the instantaneous frequency of the detected peaks, and adapting the equal tempered reference frequencies iteratively until the distance between detected and reference frequencies is minimal. The adaptation amount is calcucated by the lowpass filtered geometric mean of previous and current reference frequency estimate.

Zhu et al. [6] computed a constant Q transform (CQT) with the frequency spacing of $10\,cent$ over a range of 7 octaves. The detected peaks in the CQT spectrum are grouped based on the modulus distance against the concert pitch. If the maximum energy of the resulting 10-dimensional vector is above a certain energy threshold, it is used for later processing. For the results of all processing blocks (if not discarded), a 10-dimensional so-called tuning pitch histogram is computed, and the tuning frequency is chosen corresponding to the bin with the maximal count.

Using a CQT with $33\,cent$ frequency spacing, Harte and Sandler [7] estimate the exact peak positions by interpolation. A histogram of the peak positions based on the modulus distance against the concert pitch is computed over the length of the audio file, and the tuning frequency is set according to its maximum.

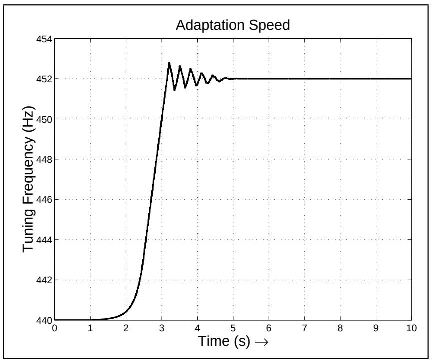

Figure 1. Adaptation of tuning frequency from initial setting of $440\,Hz$ to target $452\,Hz$

In the context of single-voiced input signals, Ryynänen [8] added the modulus distance of detected base frequencies to a 10-dimensional histogram that is lowpass-filtered over time. Then, a 'histogram mass centre' is computed and the tuning frequency is adjusted according to this mass centre.

4.2. Algorithm used

The method for the automatic detection of the tuning frequency used in this paper is described below. A previous version of this algorithm has been published in [9].

4.2.1. Description

The input audio samples are processed by a filter bank of steep resonance filters. In the range of 2 octaves around A4, there are 24 groups of filters in (equal tempered) halftone distance, with each group consisting of 3 filters. The mid frequencies of each group are spaced with $12\,cent$ and the mid frequency of the centered filter is selected based on the current tuning frequency assumption. All filters have constant Q. The filter output energy per processing block of length $20\,ms$ is then grouped based on the modulus distance against the concert pitch, resulting in a 3-dimensional vector E for each block n.

The symmetry of the distribution of the three accumulated energies gives an estimate on the deviation from the current tuning frequency compared to the assumption. If the distribution is symmetric, e.g. $E(0, n)$ equals $E(2, n)$, the assumption was correct. In the other case, all filter mid frequencies are adjusted with the objective to symmetrize the energy distribution in the following processing blocks. The RPROP-algorithm [10] is used as adaptation rule because it allows fast and robust adaptation without the requirement of specifically controlling the adaption step size. The adaption rule for the adjustment of the assumed tuning frequency f_{A4} of the following processing block $n + 1$ is:

$$f_{A4}(n+1) = \left(1 + \eta \cdot sign\left(E(2,n) - E(0,n)\right)\right) \cdot f_{A4}(n) \quad (1)$$

with η being scaled up if $sign$ returns the same result as for the previous block, and scaled down otherwise. To ensure high accuracy, η is initialized with a small value.

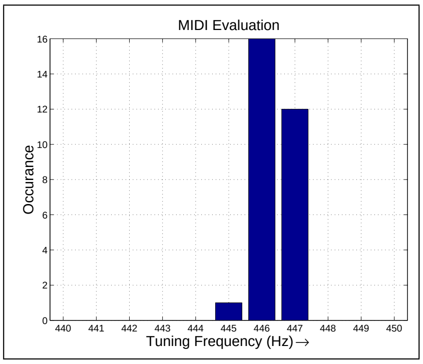

Figure 2. Distribution of results for the MIDI-generated test set tuned at $446\,Hz$

Figure 1 shows the adaptation from the initial tuning frequency $440\,Hz$ to the real frequency $452\,Hz$. Adaptation is parametrized for accuracy rather than speed in this case, so it takes the algorithm more than $3s$ to converge to the target frequency.

While this approach allows real-time processing and permanent adaptation to possibly varying tuning frequencies, in the current context the overall tuning frequency is computed by finding the maximum count in a histogram containing the estimates of all processing blocks. The histogram classes are spaced by one Hertz; while this is not completely consistent since, on the pitch scale, the width of the classes decreases slightly with increasing tuning frequency, it nevertheless was chosen considering that on the one hand the deviations are small compared to the expected accuracy, on the other hand these class labels are the most transparent for the user when interpreting the result.

4.2.2. Evaluation

To verify the algorithm's accuracy, a test with a small database of 29 input files generated from MIDI content was performed. The files were generated with equal temperament and pitched to a tuning frequency of $446\,Hz$ and were significantly longer than $10s$.

Figure 2 shows the result for this test set. The result is correct in a range of $\pm 1\,Hz$ around the reference. Coincidently, this range roughly corresponds to the just noticeable frequency difference humans are able to recognize $(2 - 4\,cent)$ [11].

The algorithm is expected to give slightly less accurate results when alternative temperaments are used.

4.3. Analysis

Processing a small database of 60 pop and 12 classical pieces, Zhu et al. [6] found that the majority of pieces are tuned to the standard tuning frequency $\pm 10\,cent$, while three pieces of this database had about $50\,cent$ deviation from the standard tuning frequency.

Here, a larger database consisting of classical music is evaluated to allow quantitative statements about tuning frequency deviations.

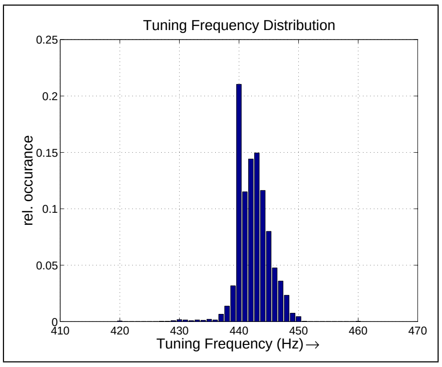

Figure 3. Distribution of results for the complete data base

4.3.1. Test sequences

The test database is a private collection of classical music, where the term *classical* is interpreted as "non-popular" music. It consists of about 300 CDs with overall 3336 tracks, and has an overall playing time of approximately 291 hours. It includes various instrumentations and ensemble sizes from solo chamber music to oratorio and integrates music from different eras of the western music history with a focus on the classic and romantic periods. The signals have CD-quality and the average track length is around 314s.

4.3.2. Results

Figure 3 shows the distribution of the detected tuning frequency per track for the whole database. While the maximum of the detected tuning frequencies can be found at frequency $440\,Hz$; the maximum itself consists of about 21% of the test database. The result's mean value is at frequency $442.38\,Hz$ with a standard deviation of $2.75\,Hz$. 95% of the results are in the range from $439 - 448\,Hz$ and only 50% of the results have tuning frequencies between $440 - 443\,Hz$. The percentage of files below $439\,Hz$ is about 3.3%.

When the results are sorted into classes roughly corresponding to the date of composition, there are no significant differences from the overall result, although the maximum of three classes can be found at higher frequencies than $440\,Hz$. It is basically not surprising that the result is similar between the classes, since many of the recordings were made at the end of the twentieth century with contemporary instruments. In further evaluations, it might be interesting to see if there are differences between classes if sorted by instrumentation and/or recording date.

The workload produced by the software is, scaled to a x86 CPU frequency of $1\,GHz$, about 6%.

5. Conclusions

While the maximum of the distribution of tuning frequencies for the test database is indeed at the standard tuning frequency $440\,Hz$, the results indicate a relatively wide frequency interval of tuning frequencies from $439 - 448\,Hz$, corresponding to a deviation from the standard tuning frequency of $-3.9\,cent$ to $31.2\,cent$.

Such a deviation is well within a detection range of $\pm 50\,cent$ per pitch; however, in addition to other deviations that cannot be influenced by the developer like temperament-based pitch frequency deviations, it may lead to (avoidable) pitch detection errors.

Thus, at least in the context of classical music, the robustness of pitch-based systems for music content analysis could most likely be improved by the usage of an automatic tuning frequency detection. Probably, similar results can be found for other musical genres that are played with acoustic instruments.

For result verification, the used software for automatic tuning frequency estimation is available online as a FEAPI plugin [12] at http://www.zplane.de/FEAPI.

References

[1] ISO, "Acoustics – standard tuning frequency (standard musical pitch)," 16:1975, ISO, 1975.

[2] Ermanno Briner, *Musikinstrumentenführer*, Philipp Reclam jun., Stuttgart, 3 edition, 1998.

[3] Mark A. Bartsch and Gregory H. Wakefield, "To Catch a Chorus: Using Chroma-Based Representations for Audio Thumbnailing," in *Proc. of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, October 2001.

[4] Eric D. Scheirer, "Extracting Expressive Performance Information from Recorded Music," Master's thesis, Massachusetts Institute of Technology, September 1995.

[5] Simon Dixon, "A Dynamic Modelling Approach to Music Recognition," in *Proc. of the International Computer Music Conference (ICMC)*, Hong Kong, August 1996.

[6] Yongweil Zhu, Mohan S. Kankanhalli, and Sheng Gao, "Music Key Detection for Musical Audio," in *Proc. of the 11th International Multimedia Modelling Conference*, Melbourne, January 2005.

[7] Christopher A. Harte and Mark B. Sandler, "Automatic Chord Identification Using a Quantised Chromagram," in *Proc. of the 118th AES Convention*, Barcelona, May 2005, number 6412.

[8] Matti Ryynänen, "Probabilistic Modelling of Note Events in the Transcription of Monophonic Melodies," Master's thesis, Tampere University of Technology, March 2004.

[9] Alexander Lerch, "Ein Ansatz zur automatischen Erkennung der Tonart in Musikdateien," in *Proc. of the VDT International Audio Convention (23. Tonmeistertagung)*, Leipzig, November 2004.

[10] Martin Riedmiller and Heinrich Braun, "A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm," in *Proc. of the IEEE International Conference on Neural Networks*, San Francisco, March/April 1993.

[11] Eberhard Zwicker and Hugo Fastl, *Psychoacoustics. Facts and Models*, Springer, 2 edition, 1999.

[12] Alexander Lerch, Gunnar Eisenberg, and Koen Tanghe, "FEAPI: A Low Level Feature Extraction Plugin API," in *Proc. of 8th Int Conference on Digital Audio Effects (DAFx'05)*, Madrid, September 2005.