

Real-time monophonic guitar pitch estimation

The feasibility of Fourier transform based methods

Luc de Jonckheere

October 14, 2021

Abstract

Short summary.

1 Introduction

Pitch estimation, which is also referred to as f_0 estimation, is an important subtask within the field of Automatic Music Transcription (AMT). The goal of pitch estimation is to estimate the pitch or fundamental frequency f_0 of a given signal. In the context of AMT, pitch estimation is used to determine what note is played in a given signal.

Real-time pitch estimation is a subproblem where we want to estimate the note associated with the measured pitch while the musician is playing it with minimal latency. This entails we have to use the latest received signal. In contrast to non-real-time methods, we have no knowledge of what may happen ahead of time and signal corresponding to previous notes is irrelevant. This limits the methods we can use to solve this problem.

If pitch estimation can accurately be performed in real-time, it can be used to create a digital (MIDI) instrument from an acoustic instrument. This digital instrument can then be used as an input for audio synthesizers, allowing musicians to produce sounds from a wide variety of instruments. Furthermore, accurate real-time pitch estimation can be used to automatically correct detuned instruments by pitch shifting the original signal to the closest harmonious note.

The Fourier transform is often used to decompose a signal into the frequencies that make up the signal. Predominant frequencies in the signal show up as spectral peaks in the frequency domain. These peaks are important to human perception of melody [5]. Other popular methods used for pitch estimation include non-negative matrix factorization, statistical model based estimation and hidden Markov model based estimation.

Our research focusses on monophonic pitch estimation. Here, we assume that the signal contains at most one note. It is much easier to perform monophonic pitch estimation compared to polyphonic pitch estimation [6], especially when using Fourier transform based methods, as fundamental limits of the Fourier transform inhibit our ability to discern two low pitched notes [3]. Furthermore, hexaphonic guitar pickups are becoming more widespread, which allows us to view the guitar as six monophonic instruments instead of one six-way polyphonic instrument. Commercial guitar synthesizer solutions from companies with big research departments such as Roland also use these hexaphonic pick-ups, which indicates the infeasibility of accurate and responsive polyphonic pitch estimation of a guitar.

This thesis builds upon a preliminary research project [4]. In our research project, we found that Fourier transform based pitch estimation methods might not be well suited for real-time use due to fundamental limitations of the Fourier transform. In this work, we will further research if Fourier transform based methods are viable, as real-time transcription research often relies Fourier transform based methods.

The goal of this thesis is to research the limits of Fourier transform based real-time pitch estimation. To correctly assess the limits, we develop a pitch estimation framework. This framework will focus on extensibility and the ability to perform automated tests. This is important, as much work in this field does not provide its associated source code. This limits the ability to build on other's work and hinders direct comparisons between different methods. Our framework can provide a common ground for the different methods and algorithms to be implemented and compared in. The framework is available at www.github.com/lucmans/digistring

2 Related work

Other papers. Sample citations [8] [11] [1].

3 Preliminaries

Jargon required to understand this paper.

3.1 Audio processing

Audio is represented by samples. We get these samples from the OS audio driver. Driver latency.

3.2 Real-time

What we mean with real-time vs how other work uses real-time. Formal vs common definition.

Real-time is a difficult concept, as it has many definitions. Here are a few examples of relevant definitions:

1. Being synced with actual clock time (or wall time). This is for instance relevant when playing media such as audio and video. When such media is played at an incorrect speed, it could be considered distorted. The hardware which keeps track of the clock time is called a real-time clock.
2. A system must response within a specified time constraint, which is called the real-time constraint or deadline. This constraint is usually a relatively short time. This definition comes from real-time computing and is relevant when making car airbags or airplane control systems. Failing to response within the real-time constraint leads to failures of the system. Real-time systems are often classified into hard, firm and soft real-time based on the impact of missing the deadline [9]. Note that system that have to respond seemingly instantaneous, such as graphical user interfaces or instant chatting/calling, are not real-time. There are no hard deadlines which the system has to respond within and the system does not fail if some delay does occur. Only user experience is slightly impacted.
3. **A system that can process data faster than it acquires data.** This is technically not real-time, however, it is often used as such in academic literature. It is important for real-time system to process data faster than it acquires it so it does not start to lag behind, however, this is an implicit deadline. Not having this deadline explicit may lead to non-sensible expectations of the system.

Even though the first definition is very relevant when working with audio, it is not relevant for us. The audio drivers of operating systems handle all timing for us. We simply have to wait for samples to be recorded and made available to our program. Moreover, when

playing back audio, we only have to keep the sampling rate in mind. The operating system handles the correct timing of the samples to our audio hardware.

In order to allow guitarists to use their guitar as a MIDI instrument, our system has to respond within a small time frame. On top of that, if the system fails to respond quickly enough, the usefulness of the result degrades, as timing is very important when playing an instrument. These restrictions would classify our system as a soft real-time system. We choose a real-time of constraint of TODO milliseconds. We elaborate on this choice in section 4.1.

Other work in real-time pitch estimation... TODO

3.3 Fourier transform

Emphasis on discrete FFT (where most papers only mention continuous, which causes misconceptions). Effect of sampling rate (which is often chosen too low) and frame size. Sensitivity for specific frequencies. Quadratic interpolation.

3.4 Music notation/theory

In modern western music, we use the twelve-tone equal temperament (12-TET) music system. This system divides an octave, which is the interval between a pitch and another pitch with double the frequency, into twelve equally spaced semitones on the logarithmic scale. The logarithmic scale is used such that the perceived interval between two adjacent notes is constant [10]. Moreover, the ratio between two frequencies in an n -semitone interval is $\sqrt[12]{2}^n$ or $2^{\frac{n}{12}}$, invariant to pitch.

Using scientific pitch notation, every note can be uniquely identified by combining the traditional note names A to G (with accidentals such as \sharp and \flat) with an octave number (e.g. E_3^\flat). An octave starts at C, which means the octave number increases between B and C. This counter intuitively implies that A_3 is higher than C_3 . Note that in 12-TET, C_4^\sharp and D_4^\flat are enharmonically equivalent. The range of a typical guitar in standard tuning is from E_2 up to E_6 .

The 12-TET music system only describes the relation between two notes in an interval. In order to play with other musicians in harmony, an arbitrary note has to be tuned to a specific frequency. Per ISO 16, the standard tuning frequency of the A_4 is 440 Hz within an accuracy of 0.5 Hz [2].

3.5 Envelope and transients

The perceived loudness of a note over time can be described using an ADSR envelope. The ADSR en-

velope of a played note is the convex hull of the waveform of the signal, see Figure 1 for an example. This convex hull can be divided in four parts: Attack, Decay, Sustain, Release. When a note is strummed on the guitar, a percussive sound is generated which causes a loud and sharp attack along with the note. This percussive sound quickly decays and only the actually fretted note will sustain. Finally, when the note is released, it dies out quickly.

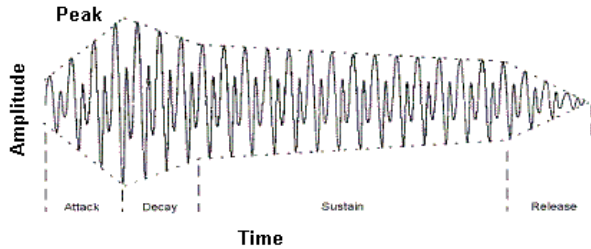


Figure 1: Example of an ADSR envelope (TODO: Better image)

As mentioned before, when strumming a note, a percussive sound is generated. This sound is called a transient and contains a high degree of non-periodic components. Transients appear very chaotically in the frequency domain and thus is often considered as noise. Since a transient is of high amplitude, it overshadows the note that will eventually sustain. Because of this, we cannot use the samples from a transient for pitch estimation. This in turn increases our minimum latency, as we have to wait for samples which do not contain the transient any more.

3.6 Fundamental frequency, overtones and timbre

When playing a note on an instrument, many sine waves are generated. The most notable frequency is called the fundamental frequency and determines what note is actually played. Integer multiples of the fundamental frequency can resonate and give rise to harmonic overtones [12]. In practice, these overtones are not exact integer multiples due to non-linear effect. TODO?: Overtone dissonance.

Furthermore, many other frequencies are generated along with the fundamental and its overtones. The instrument specific pattern of these frequencies, along with its envelope, is called the timbre of the instrument [7]. The timbre is what separates the sound of the same note played on two different instruments [10]. Generally, the amplitude of the timbre frequencies is low compared to the fundamental frequency and can be disregarded as noise in the frequency domain.

4 Thesis content

Actual research etc. Summary what was done in the research project that we build on. Notes on future work of research project.

4.1 Real-time constraint

Start with the factors coming into play when choosing the real-time constraint for our system (latency, played notes per second etc). Note on measured latency in Appendix A. Empirically found bounds on latency using our latency program

4.2 Software amplification

(Software representation of samples and FFT). The FFT works on floating point numbers but most audio interfaces give up to 24 bit integers... We found empirically that when amplifying the input signal in software, peaks in the frequency domain are much easier to detect. However, it has to be done carefully to prevent distortion artifacts.

5 Implementation

Details about the program I've written. Usage instructions, code choices, code structure, screenshots, expandability

6 Experiments

Note that the parameters were empirically optimized with informal experiments. Datasets. Actual experiments.

7 Conclusions

What we did in this thesis. Reflection on the performance of the system. Final reference to the source code.

8 Future work

What could still be improved/further researched.

A Measuring the latency of the AXON AX 100 mkII

Lekker meten en weten.

References

- [1] FFTW3 page about two times speed-up by omitting complex part.
http://www.fftw.org/fftw3_doc/One_002dDimensional-DFTs-of-Real-Data.html
Last accessed on 12-04-2021.
- [2] Iso 16:1975. acoustics — standard tuning frequency (standard musical pitch). 1975.
- [3] Eric J. Anderson. Limitations of short-time fourier transforms in polyphonic pitch recognition. Ph.d. qualifying project report, University of Washington, Department of Computer Science and Engineering, 1997.
- [4] Luc de Jonckheere. Real-time guitar transcription using Fourier transform based methods; a pitch estimation framework and overtone sieve algorithm. 2021.
- [5] Zhiyao Duan, Bryan Pardo, and Changshui Zhang. Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(8):2121–2133, 2010.
- [6] Paolo Nesi Fabrizio Argenti and Gianni Pantaleo. *Automatic Music Transcription: From Monophonic to Polyphonic*, pages 27–46. Springer Berlin Heidelberg, 2011.
- [7] Kristoffer Jensen. Timbre models of musical sounds. 2021. PhD thesis, University of Copenhagen.
- [8] S.S. Limaye K.A. Akant, R. Pande. Accurate monophonic pitch tracking algorithm for QBH and microtone research. *The Pacific Journal of Science and Technology*, 11(2):342–352, 2010.
- [9] Hermann Kopetz. *Real-Time Systems: Design Principles for Distributed Embedded Applications*. Springer US, 1997.
- [10] Michael J. O'Donnell. Digital sound modeling: Perceptual foundations of sound.
http://people.cs.uchicago.edu/~odonnell/Scholar/Work_in_progress/Digital_Sound_Modelling/lectnotes/node4.html
Last accessed on 04-10-2021.
- [11] K. M. M. Prabhu. *Window Functions and Their Applications in Signal Processing*. CRC Press, 2013.
- [12] R. S. Shankland and J. W. Coltman. The departure of the overtones of a vibrating wire from a true harmonic series. *The Journal of the Acoustical Society of America*, 10(3):161 ff, 1939.