A VERY LOW LATENCY PITCH TRACKER FOR AUDIO TO MIDI CONVERSION

Olivier Derrien,

Université de Toulon & CNRS LMA
Laboratoire de Mechanique et d'Acoustique
31 chemin Joseph-Aiguier, 13402 Marseille Cedex 20
derrien@lma.cnrs-mrs.fr

ABSTRACT

An algorithm for estimating the fundamental frequency of a single-pitch audio signal is described, for application to audio-to-MIDI conversion. In order to minimize latency, this method is based on the ESPRIT algorithm, together with a statistical model for partials frequencies. It is tested on real guitar recordings and compared to the YIN estimator. We show that, in this particular context, both methods exhibit a similar accuracy but the periodicity measure, used for note segmentation, is much more stable with the ESPRIT-based algorithm. This allows to significantly reduce ghost notes. This method is also able to get very close to the theoretical minimum latency, i.e. the fundamental period of the lowest observable pitch. Furthermore, it appears that fast implementations can reach a reasonable complexity and could be compatible with real-time, although this is not tested is this study.

1. INTRODUCTION

MIDI (Musical Interface for Digital Instruments) is the most widely used standard for connecting digital instruments. It specifies both the hardware interface and the data transmission protocol. It allows for instance to encode a melody as a collection of notes (note starting points, durations, pitches...) and to control a compatible synthesizer with an external interface, for instance a digital keyboard or a "wind controller" which mimics a wind instrument. However, for some instruments like guitars, designing an appropriate digital controller is difficult. Then, the original acoustic instrument can be used as a MIDI controller by adding an audio-to-MIDI converter. Such a device basically consists of a microphone that captures the acoustic signal produced by the instrument and a pitch-tracker which estimates the evolution of pitch during time. For guitars, one usually uses an under-saddle pickup for each string, connected to a series of monophonic pitch trackers, one for each string [1]. Such MIDI converters have been marketed for a few decades, but suffer from many flaws: latency, ghost notes, octave errors... The performance constantly improve, but latency still remains an issue: With an up-to-date Roland GR55 (built-in audio-to-MIDI converter and synthesizer) connected to a compatible acoustic guitar (Godin Multiac), we measured an average latency of 50 ms between the output of the under-saddle pickups and the audio output of the synthesizer (constant over the guitar frequency-range). Thus, playing a guitar synth is not easy and requires to develop specific skills.

In this study, we focus on the issue of latency for monophonic pitch tracking. We assume that pitch estimation is similar to fundamental frequency detection (noted f_0), and that the observed signal is harmonic. It appears that latency has several sources that add up. First, the "algorithmic delay", which is inherent to the pitch-detection algorithm. It corresponds to the length of the time-interval that is required for the algorithm to give an accurate estimation. This delay has a fundamental lower bound which related to the minimum f_0 value than can be detected. For a "Spanish" guitar[1], the minimum f_0 value is approximately 80 Hz, which corresponds to a minimum delay of 12.5 ms. Then, the "computational delay", which is the time required by the digital signal processor (DSP) to perform the pitch estimation. This delay can be reduced by increasing the speed of the DSP.

The issue of pitch detection is a classical problem and many algorithms have been proposed in the past decades. These methods can be roughly classified in two categories: time-domain and frequency-domain. Time-domain methods usually consist of finding a maximum of the auto-correlation function (or another similar function), while frequency-domain methods rely on a spectral analysis stage followed by a peak-picking stage. It was proved that time-domain methods are usually more efficient for real-time estimation of single-pitch [2]. Especially, the YIN algorithm, proposed by de Cheveigné et al. [3], can be considered as a reference f_0 estimator. It is based on the observation of a "cumulative mean normalized difference function", which is characterized by dips at the time-lags corresponding to the periodicity. This method is accurate, has a moderate complexity and a relatively low algorithmic delay. In [2], the delay of the full method was estimated around 30 ms for a "Spanish" guitar. However, this value is still approximately twice the theoretical minimum delay.

In this paper, we consider a new approach to reduce the algorithmic delay. Most f_0 estimators are non-parametric methods in the sense that they do not use *a priori* information about the signal. In contrast, parametric methods, which rely on a signal model, are known to be more precise when the observed signal correctly fits the model, but usually fail in the opposite case. For that reason, non-parametric methods are often considered more robust. However, audio signals coming from an under-saddle guitar pickup usually produce a quasi-harmonic sound with a very low noise, which justifies the use of a parametric method based on a sinusoidal signal model. In this study, we choose the Exponentially Damped Sinusoidal (EDS) model. The model parameters are estimated with a method derived from the ESPRIT algorithm [4]. This phase is similar to a spectral analysis and a peak-picking stage. To fulfill the pitch estimation, we use a spectral f_0 estimator inspired by the one proposed by Doval et al. [5]. Algorithms derived from ESPRIT are known for their good frequency resolution, but also have the reputation to require high computation time. However, fast algorithms have been proposed in the last decade [6, Chapter

[1] A "Spanish" guitar means a 6 string instrument tuned to the standard scale E-A-D-G-B-E. Thus, the lowest note is E2, corresponding to a fundamental frequency of 82.41 Hz.

V] which exhibit a complexity not much higher than a FFT. Thus, this method is theoretically suitable for real-time implementation, although this is not tested is this study.

This paper is organized as follows: In a first part, we consider more precisely the issue of algorithmic delay in a f_0 estimator. In a second part, we describe the proposed method. In a third part, we give results obtained from real guitar sounds both for our algorithm and for the YIN estimator, concerning pitch accuracy and delay. In the last part, we draw conclusions.

2. THE ISSUE OF ALGORITHMIC DELAY

Most f_0 estimators are frame-based methods. An input buffer of N samples is used, and the estimation of f_0 is made every a samples ($a \in \mathbb{N} \setminus \{0\}$). In other words, a sliding analysis window of N samples is used, with a hop-size a. The f_0 estimator should ideally be associated to the estimation of a "periodicity measure", i.e. whether the signal is pitched or not. A common periodicity measure is obtained by computing the energy of the periodic components in the signal, called "voiced" components in the case of a speech signal [7]. Such a periodicity measure influences the accuracy of the note segmentation process: A simple way to detect notes is to threshold the periodicity measure.

It is often believed that the algorithmic delay is equal to the window length, but this more complex. As exemplified on figure 1, the algorithmic delay corresponds to the time-interval between the beginning of a note and the last sample of the first window for which f_0 estimate is accurate (and eventually the periodicity measure is higher than the threshold). Sometimes, the algorithm returns the accurate f_0 even if the pitched signal does not "fill" the window (plotted case). Then, the delay can be shorter that N samples. Sometimes, the estimator takes some time to return the accurate f_0 and the delay can be longer that N samples.

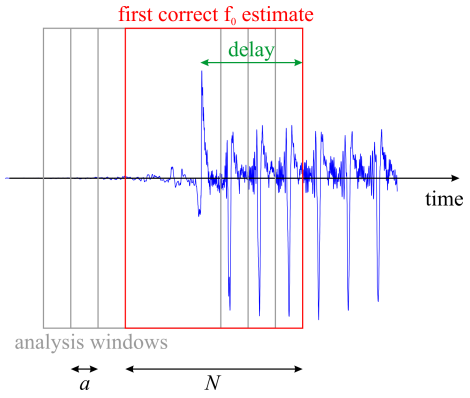

Figure 1: *Measurement of the algorithmic delay.*

As explained in [3], performing a correct f_0 estimation requires that the window length is no shorter than the largest expected period. But, according to the well known rule of thumb, a correct estimation requires enough signal to cover twice the largest period. Thus, the minimum delay is equal to $1/f_0^{\min}$, $f_0^{\min}$ being the lowest f_0 value than can be detected. As explained previously, this corresponds to approximately 12.5 ms for a "Spanish" guitar. As a consequence, the window length N must be higher that $f_s/f_0^{\min}$ where f_s is the sampling frequency. But practically, we expect a minimum delay of 25 ms.

3. THE PROPOSED METHOD

This method is divided in two stages: in the first one, the most significant sinusoidal components are extracted according to a signal model. Then, in the second stage, the most probable fundamental frequency is estimated using a statistical model.

3.1. Sinusoidal modeling

In this part, we describe the signal model and the estimation algorithm. Both have been extensively discussed in the literature. We choose to reproduce this description from a previous work [8] in order to render the paper self-contained.

In the EDS model, the signal to be analyzed is written:

$$x[n] = s[n] + w[n], \tag{1}$$

where the deterministic part $s[n]$ is a sum of K damped sinusoids:

$$s[n] = \sum_{k=0}^{K-1} \alpha_k z_k^n. \tag{2}$$

Complex amplitudes are defined as $\alpha_k = a_k\, e^{i\phi_k}$ (containing initial amplitude a_k and phase ϕ_k), and poles are defined as $z_k = e^{-d_k + 2i\pi\nu_k}$ (containing damping d_k and normalized frequency ν_k). The stochastic part $w[n]$ is a gaussian white noise.

The estimation algorithm consists in finding the best values of K, α_k and z_k for a given signal in the least square sense. In this study, an estimation algorithm proposed by Badeau *et al.* [4] is used, which is derived from the ESPRIT algorithm. The principle consists of performing an SVD on an estimate of the signal correlation matrix. The eigenvectors corresponding to the K highest eigenvalues correspond to the so-called *signal space*, while the remaining vectors correspond to the so-called *noise space*. The shift invariance property of the signal space allows a simple solution for the optimal poles values z_k. Then, the amplitudes α_k can be recovered by solving a standard least square problem. The algorithm can be described as follows:

We define the signal vector:

$$\boldsymbol{x} = \begin{bmatrix} x[0] & x[1] & \dots & x[N-1] \end{bmatrix}^T, \tag{3}$$

where N is the length of the analysis window. We assume that N is even. The Hankel signal matrix is defined as:

$$\boldsymbol{X} = \begin{bmatrix} x[0] & x[1] & \dots & x[Q-1] \\ x[1] & x[2] & \dots & x[Q] \\ \vdots & \vdots & & \vdots \\ x[R-1] & x[R] & \dots & x[N-1], \end{bmatrix}, \tag{4}$$

where $Q, R > K$ and $Q + R - 1 = N$. $Q \approx R$ was proved to be an efficient solution, thus we choose $Q = N/2$ and $R = N/2+1$. We also define the amplitude vector:

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_0 & \alpha_1 & \dots & \alpha_{K-1} \end{bmatrix}^T, \tag{5}$$

and the Vandermonde matrix of the poles:

$$\boldsymbol{Z}^N = \begin{bmatrix} 1 & 1 & \dots & 1 \\ z_0 & z_1 & \dots & z_{K-1} \\ \vdots & \vdots & \vdots & \vdots \\ z_0^{N-1} & z_1^{N-1} & \dots & z_{K-1}^{N-1} \end{bmatrix}. \tag{6}$$

Performing a SVD on $\boldsymbol{X}$ leads to:

$$\boldsymbol{X} = [\boldsymbol{U_1 U_2}] \begin{bmatrix} \boldsymbol{\Sigma_1} & 0 \\ 0 & \boldsymbol{\Sigma_2} \end{bmatrix} \begin{bmatrix} \boldsymbol{V_1} \\ \boldsymbol{V_2} \end{bmatrix}, \tag{7}$$

where $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ are diagonal matrices containing respectively the K largest singular values and the remaining singular values. $[\boldsymbol{U_1 U_2}]$ and $[\boldsymbol{V_1 V_2}]$ are respectively the corresponding left and right singular vectors. The shift-invariance property of the signal space yields to:

$$\boldsymbol{U}_1^{\downarrow} \boldsymbol{\Phi}_1 = \boldsymbol{U}_1^{\uparrow}, \qquad \boldsymbol{V}_1^{\downarrow} \boldsymbol{\Phi}_2 = \boldsymbol{V}_1^{\uparrow}, \tag{8}$$

where the poles are eigenvalues of matrix $\boldsymbol{\Phi}_1$ and $\boldsymbol{\Phi}_2$. $(.)^{\uparrow}$ and $(.)^{\downarrow}$ respectively stand for the operators that discard the first line and the last line of a matrix. Here, we estimate:

$$\boldsymbol{\Phi}_1 = (\boldsymbol{U}_1^{\downarrow})^{\dagger} \, \boldsymbol{U}_1^{\uparrow}, \tag{9}$$

where $(.)^{\dagger}$ denotes the pseudoinverse operator. The estimates of z_k are obtained by diagonalization of $\boldsymbol{\Phi}_1$. The associated Vandermonde matrix $\boldsymbol{Z}^N$ is computed. Finally, the estimates of amplitudes with respect to the least square criterion are obtained by:

$$\boldsymbol{\alpha} = (\boldsymbol{Z}^N)^{\dagger} \boldsymbol{x}. \tag{10}$$

Badeau *et al.* also proposed a criterion (ESTER) which measures the adequacy between the signal and the model [9]. It is based on the fact that equations (8) are strictly verified only when the signal exactly follows the EDS model defined by equation (2) without noise. In the general case, a distance between $\boldsymbol{U}_1^{\uparrow}$ and $\boldsymbol{U}_1^{\downarrow} \boldsymbol{\Phi}_1$ can be used to measure the model error. Is was observed that the original ESTER criterion naturally favors low values for the model order K. In order to minimize this effect, we propose a modified version of this criterion:

$$J = \frac{(K-1)^2}{\| \boldsymbol{U}_1^{\uparrow} - \boldsymbol{U}_1^{\downarrow} \boldsymbol{\Phi}_1 \|^2}. \tag{11}$$

The numerator simply performs a normalization of the denominator by the size of the matrix inside the norm. A high value for J means a good match between the signal and the model. This criterion can be used to automatically determine the best model order K, or in our case, to derive a periodicity measure.

3.2. Fundamental frequency estimation

It is assumed that each damped sinusoid in the EDS decomposition corresponds to a partial. Its frequency is related to the pole estimate by $f_k = f_s \nu_k = \frac{f_s}{2\pi} \arg(z_k)$. Doval *et al.* proposed a statistical method that allows estimating the most probable fundamental frequency of a harmonic signal given a set of partials [5]. The main idea is to compute a likelihood function of the fundamental frequency based on a probabilistic model of the observed partials. The best estimate for the fundamental frequency is the global maximum of this function. In the original method, the statistical model is elaborated and has many parameters. Estimating these parameters requires a learning database of recorded notes. Furthermore, the computation of the likelihood function can be time-consuming.

With our application, a low-complexity algorithm is desirable. We also wish that our method does not depend on a learning database. Thus, we modify the model in order to reduce the complexity and to minimize the number of parameters. In particular, the distribution of energy between partials is not modeled. This probably degrades the efficiency of the f_0 estimation compared to the original method, but it happens to be sufficient for this application.

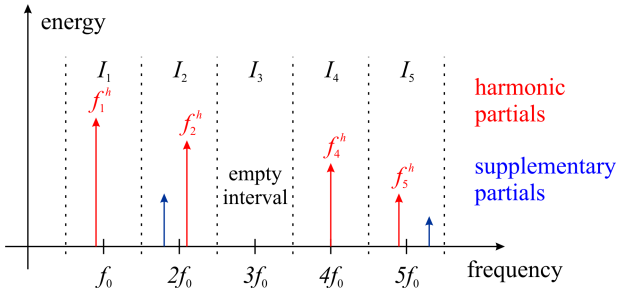

Figure 2: *Classification of partials for a given* f_0.

For a given value of f_0, we define a set of frequency intervals I_m centered on mf_0:

$$I_m = \left[\left(m - \frac{1}{2}\right) f_0, \left(m + \frac{1}{2}\right) f_0 \right], \quad m \in \mathbb{N} \setminus \{0\}, \tag{12}$$

which define a partition of the frequency scale. The partials are dispatched in these intervals according to their frequency f_k. Some intervals can contain several partials, and some others can be empty. In each non-empty interval, we define the most probable "harmonic partial" as the one which frequency is closer to mf_0, noted f_m^h. The others are called "supplementary partials" (see figure 2). The likelihood function is written as:

$$L(f_0) = \left[\prod_{m \in \mathcal{M}} g\left(\frac{f_m^h}{f_0} - m \right) \right] P_S(f_0) \, P_E(f_0), \tag{13}$$

where $\mathcal{M}$ is the set of indices m corresponding to non-empty intervals I_m. The first term is the *a posteriori* probability to observe the set of harmonic partials. The second and third terms, $P_S(f_0)$ and $P_E(f_0)$, are respectively the *a posteriori* probability to observe the set of supplementary partials and empty intervals. g is a probability function that models the frequencies of harmonic partials, which is assumed to be gaussian:

$$g\left(\frac{f_m^h}{f_0} - m \right) \propto e^{-\frac{1}{\sigma^2}\left(\frac{f_m^h}{f_0} - m\right)^2}. \tag{14}$$

σ^2 represents the variance of the reduced frequencies f_m^h / f_0 around the mean value m. $P_S(f_0)$ and $P_E(f_0)$ are estimated by:

$$P_s(f_0) = 1 - \left(\frac{N_S}{K} \right)^{\alpha_S}, \qquad P_E(f_0) = 1 - \left(\frac{N_E}{M} \right)^{\alpha_E}, \tag{15}$$

where N_S is the number of supplementary partials, N_E the number of empty intervals and M the total number of intervals. α_S and α_E are constants that allow adjusting the influence of N_S and N_E on the likelihood function.

Thus, when the frequencies of the harmonic-partials are close to mf_0, the likelihood increases. When the number of supplementary partials or empty intervals increases, the likelihood decreases. This method naturally avoids octave errors: A lower (resp. higher) octave generates supplementary partials (resp. empty intervals), which lowers the probability $P_S(f_0)$ (resp. $P_E(f_0)$) and finally lowers the likelihood. However, this requires a fine tuning on α_S and α_E.

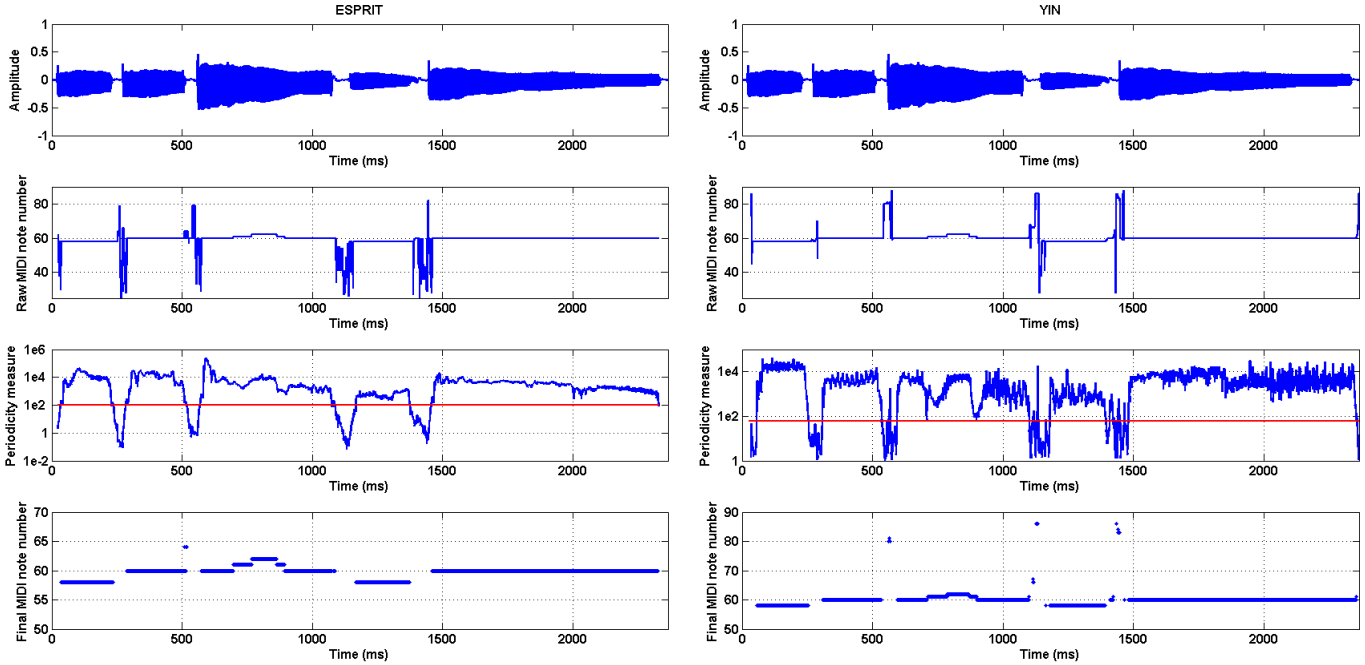

Figure 3: *Outputs of the estimation process for a short sequence of notes, ESPRIT (left column) and YIN (right column). First row: Signal waveform. Second row: MIDI note returned by the algorithm. Third row: Periodicity measure (blue) and threshold (red), log scale. Fourth row: Final MIDI note after thresholding the periodicity measure.*

3.3. Implementation details

The EDS estimation algorithm is applied on analysis segments without weighting function (also called "rectangular analysis window" in the literature). The model order K can be set to a constant, or one can define maximum and minimum values and use the ESTER criterion to select the optimal order. From our experiments on isolated guitar notes, it appears that $K = 6$ is good choice for a constant. Otherwise, K may vary between 4 and 12. Choosing a constant K saves execution time but this is sub-optimal. More precisely, bass notes usually have more harmonics than treble notes. Observing a larger set of harmonics on bass notes is desirable because these notes are more difficult to detect (there are fewer fundamental periods in the analysis window), and a larger set of harmonics gives a more robust estimation of f_0.

A periodicity measure ideally measures the energy of the periodic component in the signal. The criterion J defined in equation (11) is simply a ratio (without dimension) that measures the signal-to-model adequation. Thus, we derive our periodicity measure by multiplying J by the energy of the signal in the analysis window.

The estimation of f_0 implies computing the likelihood for all possible values of f_0. This can be accelerated by testing only discrete values, for instance on the tempered scale. If a finer estimation is required, a refinement stage can be added [5]. We set $\sigma = 1/8$, $\alpha_S = 8$ and $\alpha_E = 4$. This set of parameters appears to give a robust estimation over all the guitar frequency range. However, it is possible to choose a different set of parameters for each f_0 which could improve the detection accuracy.

Concerning complexity, ESPRIT is obviously the most critical part. A non-optimized version of ESPRIT has a complexity in $O(N^3)$ which is hardly suitable for real-time implementation. But a fast implementation of ESPRIT [6] has a complexity in $O(KN(K + \log(N)))$. When K is small, this reduces to $O(KN \log(N))$, which is not much more than a FFT. When the overlap between adjacent analysis windows is high, using adaptive algorithms allow to reduce again the complexity [6].

4. RESULTS AND DISCUSSION

In this section, we report test results for our algorithm and the YIN estimator on the same audio excerpts. The signal is the output of an under-saddle piezo-pickup on a solid-body acoustic guitar. The original signal is sampled at 44.1 kHz, downsampled at 11.025 kHz to reduce complexity. This appears to be sufficient for estimating the highest pitch on a Spanish guitar (between 930 and 1200 Hz). The implementation is in Matlab, and thus the estimation is an offline process. According to the results given in section 2, the minimum buffer length is $N = 138$ for $f_0^{\min} = 80$ Hz. We choose for both methods a hop-size of $a = 8$ samples, which corresponds to 0.72 ms.

For the YIN estimator, the author's implementation [10] is used. For the proposed method, the implementation of the ESPRIT-based estimator relies on the DESAM Toolbox [11], which is non-optimized. The minimum f_0 is set to 80 Hz for both methods. Buffer length was adjusted so that a correct estimation of f_0 is obtained for the whole guitar range. $N = 300$ is the minimal value for the YIN estimator, and $N = 260$ is the minimal value for the ESPRIT-based method. The YIN estimator gives a continuous frequency estimation that we round to the tempered scale. The new method is implemented only for discrete f_0 values corresponding to the tempered scale. Frequencies are then converted into MIDI note index for both methods. The periodicity measure in the case of YIN is the inverse of the so-called "aperiodicity mea-

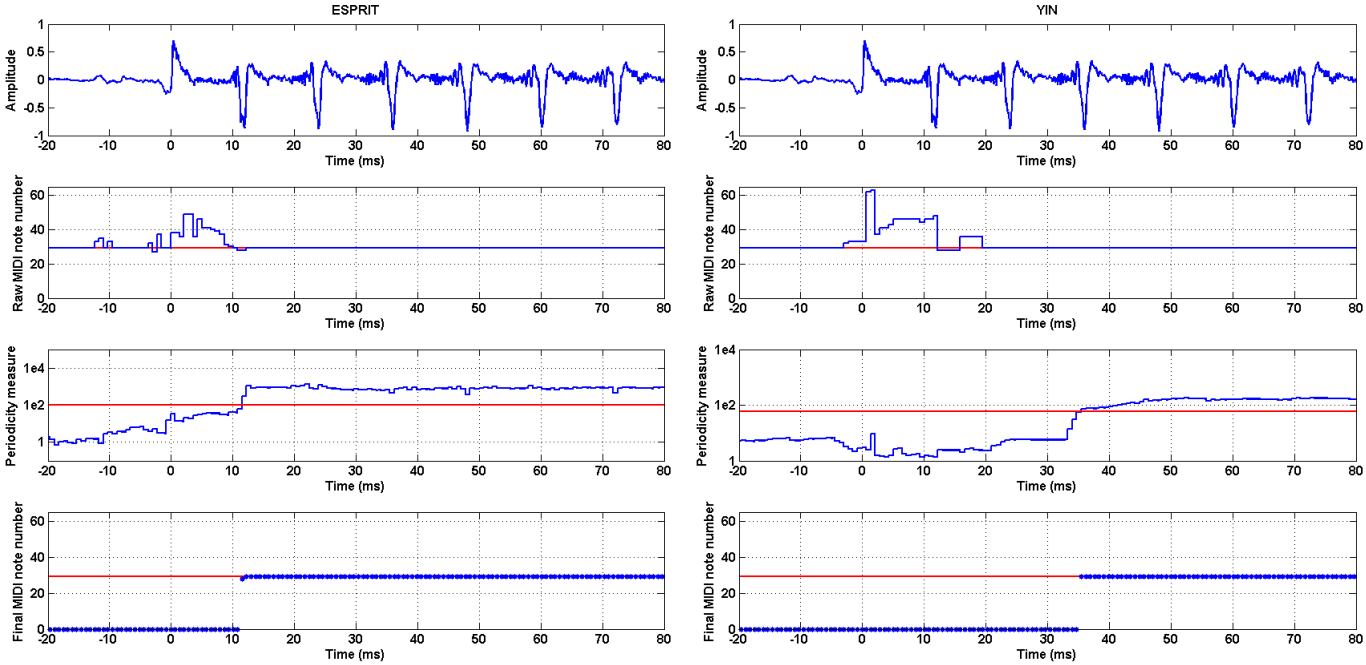

Figure 4: *Outputs of the estimation process for E2 note, ESPRIT (left column) and YIN (right column). First row: Signal waveform. Second row: MIDI note returned by the algorithm (blue) and theoretical value (red). Third row: Periodicity measure (blue) and threshold (red), log scale. Fourth row: Final MIDI note after thresholding the periodicity measure (blue) and reference value (red). Time origin is manually aligned with the onset.*

sure" returned by the algorithm [10]. For both methods, the note segmentation is obtained by thresholding the periodicity measure. The thresholds were adjusted empirically in order to get accurate segmentation on several test recordings. The threshold was set to 100 for ESPRIT and to 60 for YIN. Although, in a finely tuned application, a different threshold could be set for each string.

On figure 3, we plot the results for both algorithms on a sequence of high-pitched notes played on the same string, with a pitch-bend during the third note. The estimated MIDI note is accurate and stable for both methods when the signal is stationary. As expected, both return insignificant pitch values between the notes. The case of the periodicity measure is more contrasted: With the YIN algorithm, the periodicity evolves sharply in time, but is not very stable. One can not define a threshold on the periodicity measure that avoids ghost notes. With ESPRIT, the periodicity is more stable and an accurate thresholding can avoid most ghost notes, but it evolves more slowly in time.

On figure 4, we plot the results for a low-pitch single note (E2), which is the lowest note on a Spanish guitar, and zoom around the onset. With both methods, the estimated MIDI note is accurate and stable after a transition phase. As regards the algorithmic delay (we do not consider the computational delay in this section), the ESPRIT-based method returns the correct raw MIDI note after 12 ms, which is approximately one period of the signal (i.e. the theoretical minimum value), whereas YIN returns the correct raw MIDI note after 20 ms. However, one must take into account the periodicity measure to evaluate the actual delay. Both methods exhibit a raising front on the periodicity which allows a precise thresholding, approximately 12 ms after the onset for the ESPRIT-based method and 35 ms after the onset for the YIN algorithm. This is close to the value obtained by Knesebeck *et al.* in [2].

On figure 5, the results for a high-pitch single note (E4) are plotted. As expected, the results are globally similar to the previous case because the analysis parameters (especially the window size) did not change. However, one can see that the periodicity measure is less sharp with ESPRIT: there is a shelf between 12 and 24 ms that could extend the delay, or even generate ghost notes, depending on the threshold value. This can be explained by the fact that the signal exhibits a pseudo-periodicity before the onset that might come from the interaction between the string and the pick. The periodicity onset is sharper with YIN.

5. CONCLUSION

In this paper, an algorithm for estimating the fundamental frequency of single-pitch notes was described. The application to audio-to-MIDI conversion for guitar was especially considered. This application requires very-low algorithmic delay, which is still an issue with state-of-the-art pitch trackers. In order to minimize this delay, a new method was proposed. The first stage, equivalent to a spectral peak-picking algorithm, uses an algorithm from the literature derived form the ESPRIT method. The second stage is a fundamental frequency estimator inspired by the method proposed by Doval *et al.*, which consists in maximizing a likelihood function. The new method was tested on real guitar recordings and was compared to the YIN estimator proposed by de Cheveigné *et al.* which can be considered as a reference method. It was showed that, on this test material, both methods exhibit a similar accuracy, but it is important to notice that only the closest MIDI note was considered, and not the continuous fundamental frequency estimation. Concerning the periodicity measure which is used for

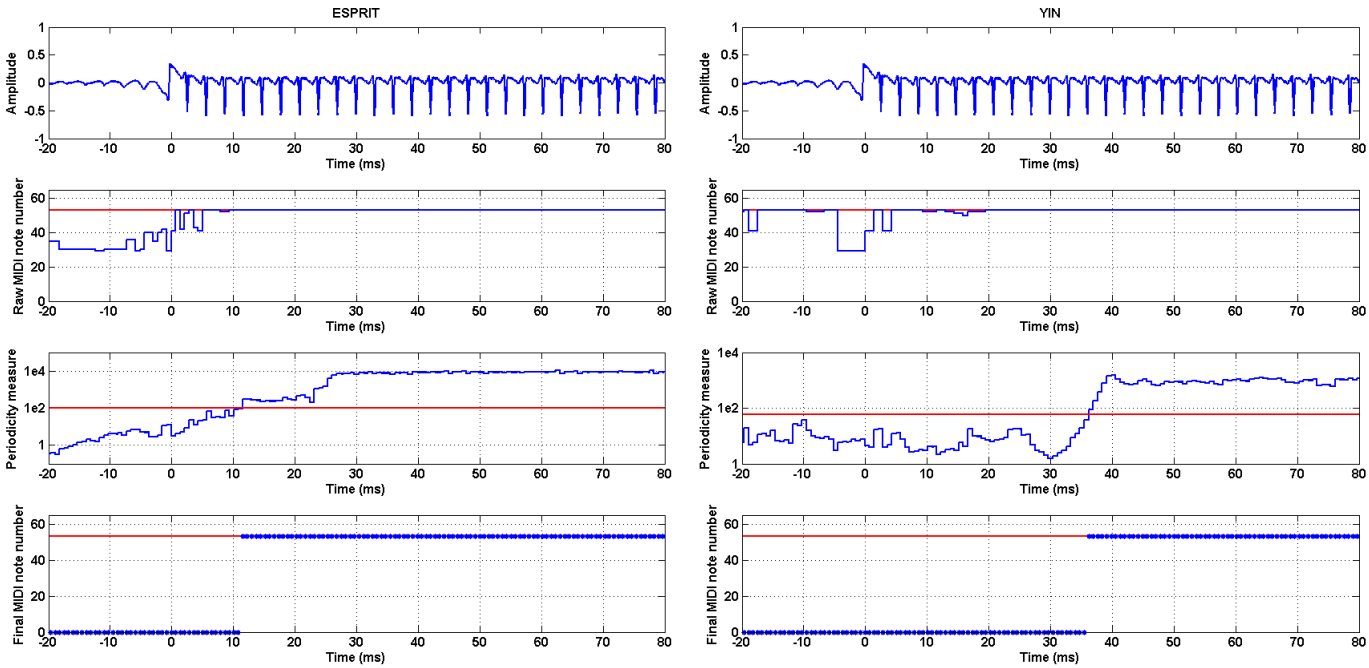

Figure 5: *Outputs of the estimation process for E4 note, ESPRIT (left column) and YIN (right column). First row: Signal waveform. Second row: MIDI note returned by the algorithm (blue) and theoretical value (red). Third row: Periodicity measure (blue) and threshold (red), log scale. Fourth row: Final MIDI note after thresholding the periodicity measure (blue) and reference value (red). Time origin is manually aligned with the onset.*

note segmentation, the new method was found more stable than the YIN estimator. This allows to significantly reduce ghost notes that are commonly observed in audio-to-MIDI conversion. It was also showed that the ESPRIT-based method is able to provide note tracking with an algorithmic delay that is very close to the theoretical limit, i.e. the fundamental period of the lowest observable pitch, which is not the case with the YIN method. For that reason, this new estimator may allow to significantly reduce the latency of audio-to-MIDI conversion. However, the issue of computational cost is crucial. The YIN estimator is a fast method, well suited for real-time implementation. In this preliminary study, our method was only tested off-line using a non-optimized implementation in Matlab. But theoretical studies have showed that fast implementations of the ESPRIT algorithm can reach a reasonable complexity in $O(KN\log(N))$ where K is the number of partials to be observed (typically 6) and N is the length of the analysis window (here less than 300 points). This is not much more than a FFT, which means that an optimized version would be theoretically compatible with real-time. This point will be investigated in the future.

6. REFERENCES

[1] U. Zölzer, Ed., *DAFX, Digital Audio Effects*, J. Wiley & Sons, New York, NY, USA, 2002.

[2] A. von dem Knesebeck and U. Zölzer, "Comparison of pitch trachers for real-time guitar effects," in *Proc. Digital Audio Effects (DAFx-10)*, Graz, Austria, Sept. 2010.

[3] A. de Cheveigné and H. Kawahara, "Yin, a fundamental frequency estimator for speech and music," *J. Acoust. Soc. Am.*, vol. 111, no. 4, pp. 1917–1930, Apr. 2002.

[4] R. Badeau, R. Boyer, and B. David, "EDS parametric modeling and tracking of audio signals," in *Proc. DAFx'02*, Hamburg, Germany, Sept. 2002.

[5] B. Doval and X. Rodet, "Estimation of fundamental frequency of musical sound signals," in *Proc. ICASSP'91*, Toronto, Ontario, Canada, May. 1991.

[6] Roland Badeau, *High resolution methods for estimating and tracking modulated sinusoids. Application to music signals.*, Ph.D. thesis, École Nationale Supérieure des Télécommunications, ENST2005E007, Paris, France, Apr. 2005, in French.

[7] G. Richard and C. d'Alessandro, "Analysis/synthesis and modification of the speech aperiodic component," *Speech Communication*, , no. 19, pp. 221–244, 1196.

[8] A. Sirdey, O. Derrien, R. Kronland-Martinet, and M. Aramaki, "Modal analysis of impact sounds with esprit in gabor frames," in *Proc. Digital Audio Effects (DAFx-11)*, Paris, France, Sept. 2011.

[9] R. Badeau, B. David, and G. Richard, "Selecting the modeling order for the esprit high resolution method: an alternative approach," in *Proc. ICASSP'04*, Montreal, Quebec, Canada, May 2004.

[10] "The YIN algorithm documentation," Available at http://mroy.chez-alice.fr/yin/index.html.

[11] "The DESAM toolbox," Available at http://www.tsi.telecom-paristech.fr/aao/en/2010/03/29/desam-toolbox-2/.