MONOPHONIC PITCH DETECTION BY EVALUATION OF INDIVIDUALLY PARAMETERIZED PHASE LOCKED LOOPS

Johannes Böhler, Udo Zölzer

Department of Signal Processing and Communications
Helmut Schmidt University
Hamburg, Germany
johannes.boehler@hsu-hh.de

ABSTRACT

This paper describes a new efficient and sample based monophonic pitch tracking approach using multiple phase locked loops (PLLs). Hereby, distinct subband signals traverse pairs of individually parameterized PLLs. Based on the relation of the instantaneous pitch sample of respective PLLs to one another, relevant features per pitch candidate are derived. These features are combined into pitch candidate scores. Pitch candidates which exhibit the maximum score per sampling instance and exceed a voicing threshold, contribute to the overall pitch track. Evaluations with up to date datasets show that the tracking performance, compared to implementations which use only one PLL has significantly improved and nearly approaches the scores of a state of the art monophonic pitch tracker.

1. INTRODUCTION

Pitch is a perceptual feature which is still subject to discussion and lacks an explicit mathematical definition. In the presented approach pitch is therefore considered to be the momentarily present fundamental frequency. In 3.2 the definition of pitch is discussed further with regard to the comparison with an alternative pitch tracking technique. If pitch information is extracted from an audio signal, it can be used to control further audio signal processing in many possible ways. Until today various monophonic pitch tracking techniques have been developed. Some of these approaches deliver robust and satisfying results. However, most of these systems employ block-based analysis and their implementation can be computational expensive. A block-based approach, named PYIN, applying the difference function paired with probabilistic evaluation and post-processing [1] yields the best results to date. This paper introduces a new sample-based approach for monophonic pitch extraction using multiple PLLs which is computationally efficient and suitable for implementations on low-power processors with limited resources. PLLs have been used for music information retrieval purposes as beat tracking [2] and monophonic pitch detection before. A pitch tracker combining numerous phase locked loops, involving a lot of redundancy leading to high computational cost, is presented in [3]. In another approach a single, modified PLL is used for pitch extraction [4]. This leads to satisfying results for input signals where the respective overtone energy is lower than the energy of the fundamental frequency. Otherwise octave errors might occur and the single PLL locks to overtone frequencies. From here on this algorithm is referenced throughout this paper as Single PLL while the algorithm which is presented in the following text will be referred to as Multi PLL. If instead of a single PLL multiple PLLs with equal parametrization are applied in differing, slightly overlapping subbands, one can observe merging pitch tracks of neighboring PLLs [5]. This observation leads to the idea to combine distinct subbands and variably configured PLLs in a new way in order to exploit the occurring concurrence of pitch tracks on periodic monophonic audio input. The following paper is structured as follows. Section 2 gives a system overview by presenting how multiple bandpass filters and phase locked loops are configured and combined. It shows how particular PLL pitch samples are interpreted in order to extract significant features regarding the instantaneous fundamental frequency of the monophonic audio signal. Based on these features the derivation of PLL-dependent pitch candidate scores and the successive selection of a candidate is described. Section 3 compares the pitch tracking performance of the approach presented in this paper with the original Single PLL pitch detector and the PYIN algorithm by means of a guitar-based dataset. Section 4 summarizes the findings of this study and discusses possible future enhancements.

2. SYSTEM OVERVIEW

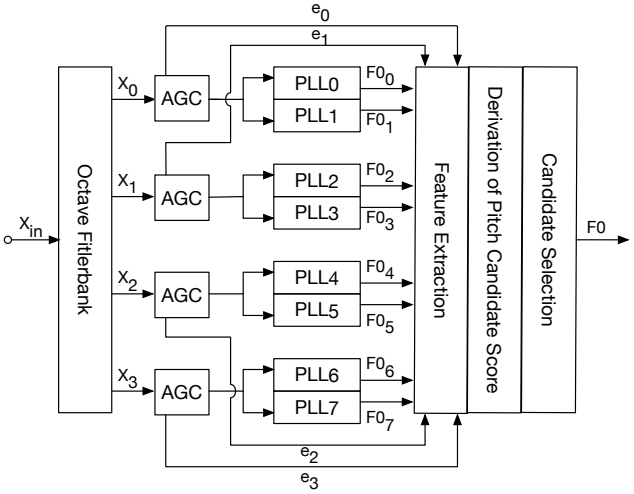

Figure 1: *Block diagram of the Multi PLL algorithm*

First, the input audio signal is filtered using a decimation filter in order to conduct a successive downsampling to a sample frequency of 11 kHz. A filter bank then divides the input audio signal into 4 octave bands. For each subband signal an envelope is calculated in order to generate a constant envelope signal of unity amplitude (AGC block in Fig. 1). This dynamic pre-processing allows the PLLs to achieve an optimal tracking performance. After

the subband signals have passed the gain control stage they are fed to the respective PLL pairs. Output samples of all 8 PLLs are interpreted in order to extract 5 features per PLL. Each feature is based on a different relation as pitch pair deviation, the number of pitch candidates assigned to relative subtones/overtones, number of close pitch candidates and pitch slope. A pitch candidate score for each PLL output sample is derived by combining these features. The pitch candidate scores reflect signal properties as periodicity, harmonicity and pitch slope. Hereby the PLL itself covers the feature extraction which is related to periodicity, while the combination of particular pitch tracks into features pursues amongst others the quantification of harmonicity. The pitch candidate with the highest score is selected and contributes to the overall pitch track F0, provided that a certain voicing threshold is exceeded.

2.1. Filterbank

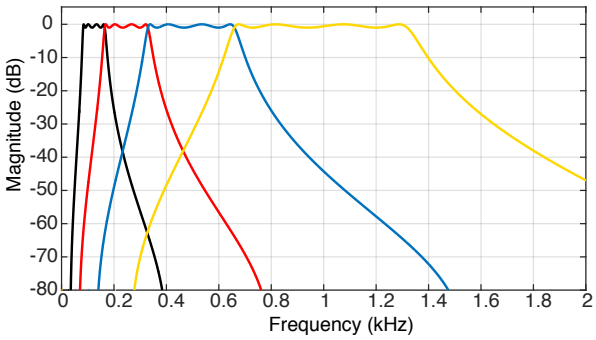

Figure 2: *Absolute frequency responses of utilized 8th order elliptic filters*

The filter bank is composed of $I = 4$ bandpass filters as depicted in Fig. 2. Each sub band channel $i \in [0, \ldots, I-1]$ spans over a region of one octave. 8th order elliptic bandpass filters with a stop band attenuation of 80 dB and a passband ripple of 1 dB are used to facilitate a sufficient edge steepness. This parametrization results in a 63 dB/Octave roll-off, which supports the isolation of fundamentals as illustrated in Fig. 4. The subbands are enclosed by the ripple frequencies

$$f^r_{i,j} = 80.06 \text{ Hz} \cdot 2^{i+j}, \tag{1}$$

where $j = 0$ denotes the lower ripple frequency and $j = 1$ denotes the upper ripple frequency. The magnitude frequency responses $|H_i(e^{j\omega})|$ of the filters are shown in Fig. 2. By filtering the input signal in both the forward and the reverse direction, a phase distortion of the output signal can be prohibited.

The approach described in this study mainly aims at providing the best possible tracking performance of the overall system under ideal performance of the subsystems. Ideal performance with regard to the filterbank subsystem means, the attainment of a certain edge steepness of respective subband filters without the emergence of phase distortions. This ensures the isolation of the fundamental frequency without the presence of the first overtone in the target octave band. At the same time a frequency dependent delay of pitch tracks is prohibited. The above mentioned characteristics can be achieved in the simplest way if the processing of the filterbank is conducted offline. The implementation of the filterbank is the only reason why the overall system is bound to offline processing. All succeeding modules can be adapted for real-time processing without much effort and worth mentioning side effects. In order to implement the whole system in a real-time-capable fashion an alternative design for the filterbank has to be considered.

Within each channel a temporal envelope $e_i(n)$ is generated whose inverse value is used to generate the constant envelope signal

$$\bar{x}_i(n) = \frac{x_i(n)}{e_i(n) + e_{min}}. \tag{2}$$

$e_i(n)$ is computed using the smoothed decoupled peak detection algorithm [6] with attack $\tau_a = 50$ ms and release $\tau_r = 100$ ms. In the denominator e_{min} is added as a constant offset in order to prevent divisions by zero. The chosen parametrization depicts a trade off between minimum-time envelope tracking and the containment of arising nonlinearities within the subband. The further usage of $e_i(n)$ for feature extraction is described in 2.3.

2.2. Phase locked loop

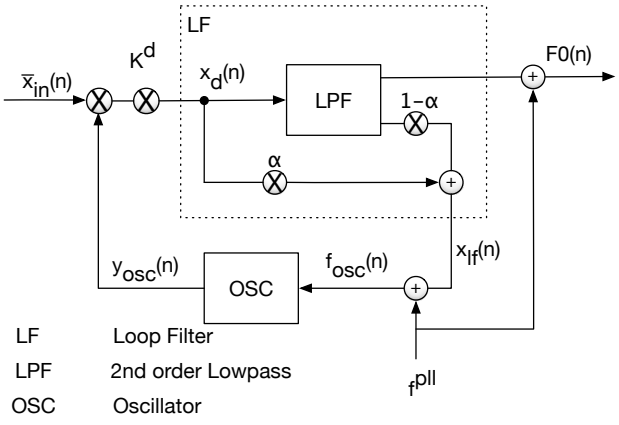

Figure 3: *Block diagram of the modified PLL*

In order to track partials which are present in a respective subband, two third order PLLs operating in a nonlinear mode are installed. The phase detector is implemented as a simple multiplier which uses the constant envelope signal $\bar{x}_{in}(n)$ and the oscillator output $y_{osc}(n)$ as input signals (Fig. 3). The resulting signal contains the difference-frequency and the sum-frequency of the real valued input signals. This signal is in succession amplified by the constant gain factor K^d which controls the frequency range of detection and the sensitivity of the PLL towards changes of the phase difference between the two signals. $x_d(n)$ then traverses a 2nd order lowpass filter (LPF in Fig. 3) with a cutoff frequency of 23 Hz, which is also part of the loop filter, in order to eliminate the sum frequency component of the multiplier output. The forward path is continued by adding the constant PLL specific center frequency f^{pll} to the lowpass filter output. The resulting output signal denotes the F0 track of the PLL. This F0 track is then filtered irrespective of the loop filter in the forward and reverse direction using a first order recursive moving average filter with non recursive coefficient 0.05 and recursive coefficients [1, -0.95] in order to further attenuate undesired high frequency oscillations.

In known implementations the lowpass filter (LPF in Fig. 3) alone represents the loop filter in the feedback path. Because

this implementation requires an immediate tracking of varying frequencies over a big range without the presence of a constant carrier frequency [7], modifications to the loop filter have to be applied in order to obtain satisfying tracking results. This modification is realized using a low frequency shelving filter in the feedback path formed by the weighted sum of x_d(n) and the lowpass-filtered phase detector output [4]. For α a value of 0.35 is chosen. In the feedback path

$$f_{osc}(n) = f^{pll} + x_{lf}(n) \tag{3}$$

controls the instantaneous frequency of the oscillator according to which the phase is incremented. The oscillator emits the real-valued signal

$$y_{osc}(n) = \cos\big(\phi(n)\big) \tag{4}$$

with the wrapped phase

$$\phi(n) = \left(\phi(n-1) + 2\pi\frac{f_{osc}}{f_s}\right) \bmod 2\pi. \tag{5}$$

By feeding $y_{osc}(n)$ and $x_{in}(n)$ to the multiplier, the loop is closed. The basic concept behind the pairwise positioning of the PLLs is the idea that a fundamental frequency within one subband must be tracked by both, the upper and the lower PLL as illustrated in Fig. 4. The parametrization of the 8 PLLs differs in the used values for the gain of the phase detector output K^d, and the PLL center frequency f^{pll}. The gain of the phase detector

$$K_i^d = 600 \cdot (i+1) \tag{6}$$

is adapted per band in a linear fashion and has been determined experimentally. Each subband i is enclosed by a respective PLL pair with lower

$$f_{i,l}^{pll} = 80.06 \text{ Hz} \cdot 2^{i-\frac{1}{12}} \tag{7}$$

and upper

$$f_{i,u}^{pll} = 80.06 \text{ Hz} \cdot 2^{i+\frac{13}{12}} \tag{8}$$

center frequency. For the following steps the octave indexes

$$i \rightarrow p = 2i + j \tag{9}$$

are mapped to PLL indexes p where $j = 0$ refers to the lower and $j = 1$ to the upper PLL.

PLL pairs assigned to subbands that contain merely overtone energy tend to track distinct overtones, while PLL pairs that track the fundamental coincide. If there is no sufficient periodic signal portion apparent in the respective octave band, the F0 track falls back to the PLLs center frequency. This behavior, which supports the differentiation between the fundamental and higher order partials, is shown in Fig. 5. The same principle applied to a guitar lick recording is depicted in Fig. 6. The spectrogram is overlayed with candidate pitch tracks.

2.3. Feature Extraction

For every sampling instance a single pitch sample for each of the 8 individually parameterized PLLs ($F0_p(n)$) is emitted. Based on these samples, the following 5 features are extracted and described in detail. Each feature is determined by the mapping of a calculated difference Δ or a count of pitch candidates N that fulfill certain constraints.

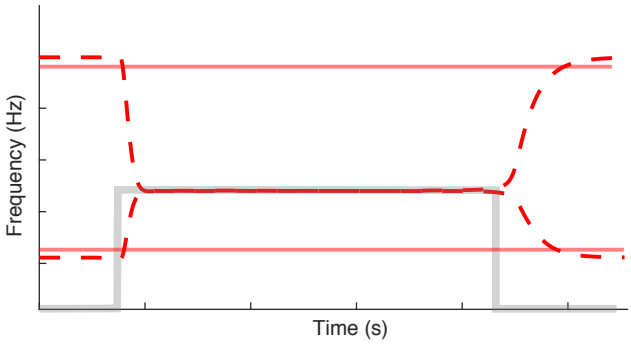

Figure 4: *The PLL pair that corresponds to the channel in which the fundamental is contained concurs (dashed lines). The continuous lines depict the ripple frequencies of the elliptic bandpass filter and the grey line represents the ground truth.*

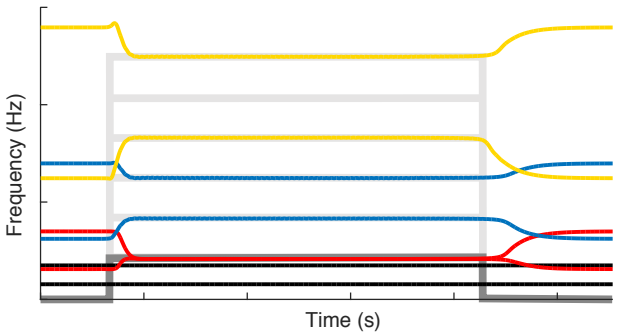

Figure 5: *Optimal tracking of the fundamental and corresponding overtones. Only the PLL pair that tracks the fundamental coincides (red), while all other PLLs remain on their center frequency (black) or track distinct overtones (blue, yellow). The grey lines denote the partial frequencies.*

2.3.1. Pitch pair deviation

We assume that PLL pairs which process the same subband signal coincide if the fundamental is contained within this channel (Fig. 4). In this context

$$\Delta f_p(n) = \left|1200 \cdot \log_2\left(\frac{F0_{\lfloor \frac{p}{2} \rfloor \cdot 2}(n)}{F0_{\lfloor \frac{p}{2} \rfloor \cdot 2+1}(n)}\right)\right| \tag{10}$$

quantifies the distance between $F0_{\lfloor \frac{p}{2} \rfloor \cdot 2}(n)$ and $F0_{\lfloor \frac{p}{2} \rfloor \cdot 2+1}(n)$, the particular pitch pair residing in subband $i = \lfloor \frac{p}{2} \rfloor$, on a logarithmic scale. This leads to the feature

$$F_p^{\Delta f}(n) = \frac{100 - \Delta f_p(n)}{100} \tag{11}$$

with $-\infty < F_p^{\Delta f}(n) \leq 1$. The negative range of $F_p^{\Delta f}(n)$ expresses the importance of coinciding pitch pairs when seeking the fundamental frequency.

2.3.2. Pitch candidates assigned to relative subtones

For a PLL pitch sample $F0_p(n)$ which is assigned to a fundamental, no other pitch candidate $F0_q(n)$ may refer to a relative subtone. There are two criteria a potential subtone $F0_q(n)$ of $F0_p(n)$ has to fulfill in order to be classified accordingly. The absolute difference in Cent between $F0_p(n)$ and an integer multiple of $F0_q(n)$ must fall below ΔC, and the relative subband energy

$$E_i(n) = \frac{e_i(n)^2}{\sum_{j=0}^{3} e_j(n)^2}, \tag{12}$$

must exceed 3%. ΔC is set to 40 Cent in order to consider inherent oscillations of the F0 tracks as well as inaccuracies and inconsistencies of the particular F0 tracks which can occur during the attack of a tone. The subband energy is considered before the signal passes the automatic gain control stage (AGC block in Fig. 1) in order to prohibit false subtone detections caused by the amplification of noise components. A false subtone detection would lead to an incorrect exclusion of pitch candidate $F0_p(n)$ in the following scoring. The number of PLL pitch samples $F0_q(n)$ that refer to a relative subtone of a particular PLL pitch sample $F0_p(n)$ is defined as

$$N_p^{st}(n) = \sum_{q=0}^{p-1} \sum_{o=2}^{8} \begin{cases} 1 & \left| 1200 \cdot \log_2 \left(\frac{F0_p(n)}{F0_q(n) \cdot o} \right) \right| < \Delta C \wedge E_{\lfloor \frac{q}{2} \rfloor}(n) > 3\% \\ 0 & \text{else.} \end{cases} \tag{13}$$

$N_p^{st}(n)$ is mapped to the mandatory feature

$$F_p^{st}(n) = \begin{cases} 1 & N_p^{st}(n) = 0 \\ 0 & \text{else.} \end{cases} \tag{14}$$

If a relative subtone of a pitch candidate $F0_p(n)$ has been identified, its overall score is set to zero as noted in Eq. (21).

2.3.3. Pitch candidates assigned to relative overtones

Pitch candidates $F0_q(n)$ which reside near integer multiples of the currently examined PLLs pitch sample $F0_p(n)$ reinforce the assumption that $F0_p(n)$ is a fundamental frequency. The number of pitch candidates $F0_q(n)$ which are classified as overtones of $F0_p(n)$ is defined as

$$N_p^{ot}(n) = \sum_{q=p+1}^{7} \sum_{o=2}^{8} \begin{cases} 1 & \left| 1200 \cdot \log_2 \left(\frac{F0_p(n) \cdot o}{F0_q(n)} \right) \right| < \Delta C \\ 0 & \text{else,} \end{cases} \tag{15}$$

from which the feature

$$F_p^{ot}(n) = \frac{N_p^{ot}(n)}{7} \tag{16}$$

is derived. This feature favors PLLs with lower-order indexes in order to prevent overtone errors.

2.3.4. Close pitch candidates

A high number of PLL pitch samples $F0_q(n)$ that reside near pitch sample $F0_p(n)$

$$N_p^{cp}(n) = -1 + \sum_{q=0}^{7} \begin{cases} 1 & \left| 1200 \cdot \log_2 \left(\frac{F0_p(n)}{F0_q(n)} \right) \right| < \Delta C \\ 0 & \text{else} \end{cases} \tag{17}$$

exhibits that $F0_p(n)$ comparatively carries a lot of energy. This is represented by the feature

$$F_p^{cp}(n) = \frac{N_p^{cp}(n)}{N_{max}^{cp}}, \tag{18}$$

with N_{max}^{cp} = 4 based on analyzed $N_p^{cp}(n)$ outputs.

2.3.5. Pitch slope

After the attack phase of a tone has passed, its pitch is assumed to be stable with little variation in time. Therefore, a low order

$$\Delta t_p(n) = \left| 1200 \cdot \log_2 \left(\frac{F0_p(n)}{F0(n-1)} \right) \right|, \tag{19}$$

denoting a flat pitch slope, increases the feature

$$F_p^{\Delta t}(n) = \begin{cases} \frac{3 - \Delta t_p(n)}{3} & F0(n-1) \neq 0 \wedge \frac{3-\Delta t_p(n)}{3} > 0 \\ 0 & \text{else.} \end{cases} \tag{20}$$

2.4. Pitch candidate selection

The extracted features are combined into a final pitch candidate score

$$S_p(n) = \frac{F_p^{\Delta f}(n) + F_p^{ot}(n) + F_p^{cp}(n) + F_p^{\Delta t}(n)}{4} \cdot F_p^{st}(n). \tag{21}$$

It is assumed that the correct instantaneous fundamental frequency equals at least one of the PLL output samples, provided that the signal carries enough periodic and harmonic portions. Hence, the output sample with the highest pitch candidate score is determined

$$k = \underset{p \in [0,\ldots,7]}{\arg\max}\, S_p(n) \tag{22}$$

and added to the overall pitch track F0 if the score exceeds a certain voicing threshold

$$F0(n) = \begin{cases} F0_k(n) & S_k(n) > T^v_{\lfloor \frac{k}{2} \rfloor} \\ 0 & \text{else.} \end{cases} \tag{23}$$

If none of the pitch samples exceeds the threshold, the audio signal is assumed to be unvoiced. In this case a zero is appended to the overall pitch track. Outliers caused by overtones, which appear when overtone energy is present before the energy of the fundamental, are smoothed using a nonlinear median filter of order 200. A F0 track, extracted by the Multi PLL, is depicted by the blue line in Fig. 7.

In future implementations the median filter could be replaced by a statistical model for post-processing purposes like a hidden

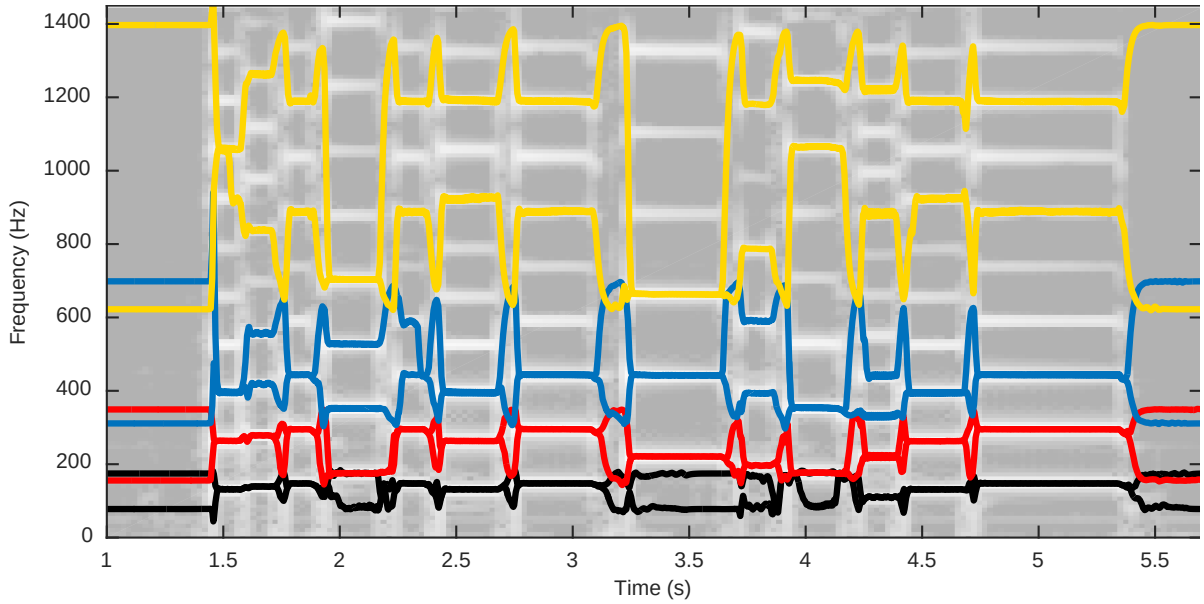

Figure 6: *Spectrogram overlayed with PLL candidate pitch tracks. Pitch tracks with identical color originate from a PLL pair and depend on the same subband signal.*

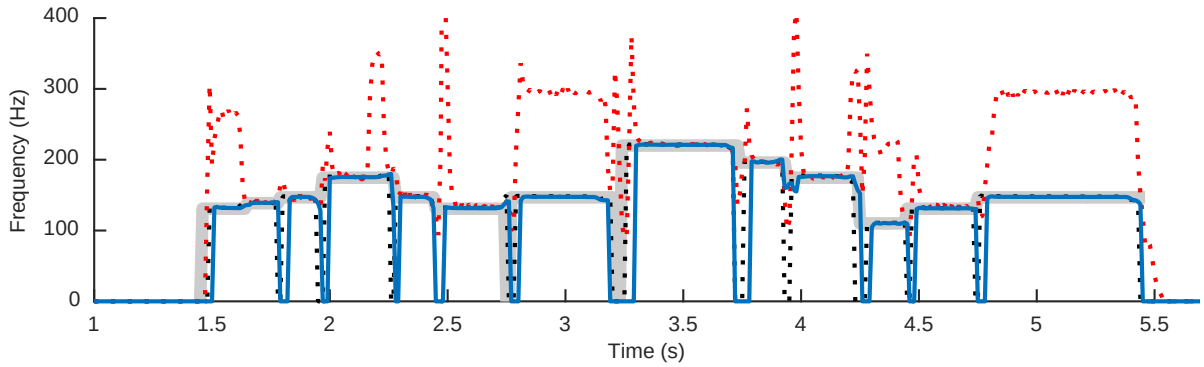

Figure 7: *Plot of the the annotated ground truth (grey), Multi PLL pitch track (blue), Single PLL pitch track (dotted red) and PYIN pitch track (dotted black).*

markov model. The first 4 extracted features could therefore be used to derive observation probabilities, while the pitch slope could be considered by given transition probabilities. In addition to that the algorithm could be optimized by an explicit weighting of the extracted features.

3. EVALUATION

The presented pitch tracker, its predecessor [4] and the PYIN [1] algorithm (with Beta distribution mean of 0.15) are evaluated using the IDMT-SMT-GUITAR dataset [1]. All monophonic guitar lick recordings (one channel RIFF WAVE format, at 44.1 kHz, 24 Bit) and corresponding annotations of dataset 2, which are tagged with playing style fingered, picked and muted as well as expression style normal, are used for the comparison. The annotations deliver pitch information wrapped in the form of note events. A note event contains data such as pitch quantized to midi notes, note onset and note offset times. According to this data, a reference pitch track (ground truth) is generated by

$$f_m = 2^{\frac{(m-69)}{12}} \cdot 440 \text{ Hz}, \tag{24}$$

in order to convert midi notes to frequency values. The tracks of all pitch detectors and the annotation are downsampled to a sampling frequency of 100 Hz. In Fig. 7 pitch tracks of the annotation and the 3 estimators are overlayed. The PYIN pitch tracker shows the fastest reaction to onset events and is capable of determining the fundamental frequency even during the attack phase of a tone. The Multi PLL algorithm avoids the overtone errors of its predecessor but exhibits delayed onset detections compared to the PYIN.

[1] `http://www.idmt.fraunhofer.de/en/business_units/smt/guitar.html`

3.1. Detection rates

Measures of binary classification as Precision and Recall cannot be applied offhand in this case. Therefore they have to be defined appropriately. In dependence on [1] we define Recall as the proportion of actually voiced samples (according to ground truth), which the extractor recognizes as voiced and tracks with a maximum deviation of ± 50 Cent. Precision is defined as the proportion of pitch samples marked by the extractor as voiced which have a maximum deviation of ± 50 Cent from reference pitch. F-Measure can be derived as the geometrical mean of Precision and Recall.

Table 1: *Detection scores for the examined pitch trackers based on the IDMT-SMT-GUITAR dataset.*

Pitch tracker	F-Measure	Precision	Recall
Multi PLL	82.63%	92.24%	75.87%
Single PLL	56.37%	55.83%	57.22%
PYIN	88.23%	90.36%	86.41%

Table 1 shows that the F-Measure of the Multi PLL has increased by approximately 26% compared to its predecessor. Especially Precision has risen by 36.41% due to less octave errors and increased pitch stability. Recall has improved by 18.65% . However, the Multi PLL algorithm misses the F-Measure of the PYIN by 5.60%. This is mainly due to a Recall which lies 10.54% below the PYINs counterpart. Solely the Precision score of the PYIN has been exceeded by 1.88%. Overall, the detection rates of the Multi PLL show that most of the errors are missing detections which are reflected in the low Recall score. These errors are mainly caused by delayed onset recognitions and undetected tones of muted licks.

The PYINs F-Measure of 88.23% for the IDMT-SMT-GUITAR dataset is lower than expected compared to other findings [1]. Recall mainly suffers from unvoiced detections for note on/off stages and muted tones. Precision is decreased by false voiced detections. Nevertheless, the PYIN algorithm depicts the best pitch tracker to date and persuades with good detection rates and exactness which is the reason why it is used as a comparative basis in the following subsection.

3.2. Exactness of pitch estimates

The examined pitch trackers should not only be able to quantize pitch to a semitone grid. If a pitch tracker is applied to audio signals emitted by vocal chords or instruments that enable intonation in between the semitone grid (e.g. fretted and fret less stringed instruments, winds etc.), it can be necessary to estimate the fundamental frequency as precise as possible. The annotations provided by the database don't qualify for the evaluation of exactness of pitch estimates due to its coarse frequency quantization. Because of this and the fact that the PYIN algorithm has already been applied to define the pitch ground truth for the Medley DB [8], its pitch track is in the following regarded as comparative basis.

In order to determine the exactness of the PLL-based pitch trackers, solely pitch samples which have been tagged as voiced and correct (in ± 50 Cent range) for all three pitch trackers are used. For each of the pitch samples of the Single PLL and the Multi PLL, the deviation in Cent to the corresponding samples of the PYIN pitch tracks is determined. Based on these deviations, the mean, the standard deviation (STD) and the median are calculated and a histogram is generated which is depicted in Fig. 8.

Table 2: *Statistical measures concerning the deviation between the PLL-based pitch trackers and PYIN.*

Pitch tracker	Mean (Cent)	STD (Cent)	Median (Cent)
Multi PLL	-2.57	7.21	-1.75
Single PLL	-2.02	16.44	-1.31

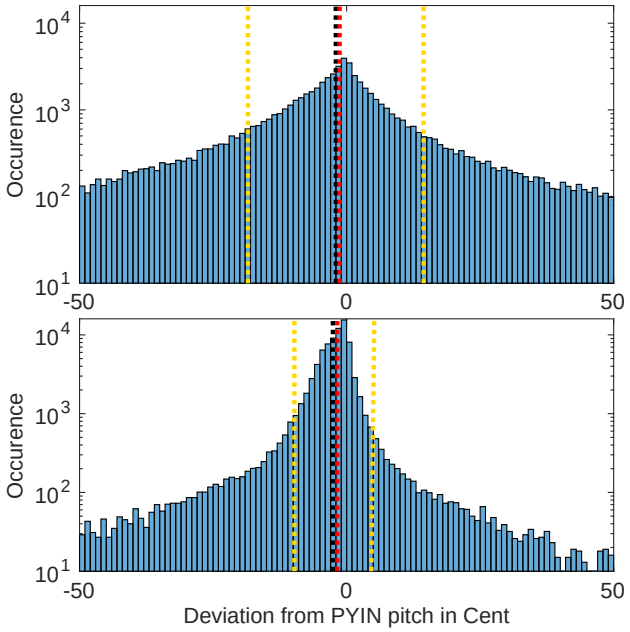

Figure 8: *Histogram of the pitch deviations in the range of ± 50 Cent between PYIN and Single PLL (top) as well as PYIN and Multi PLL (bottom). The dotted lines depict mean (black), median (red) and standard deviation (yellow).*

Table 2 reveals that the mean of the pitch deviation vector is negatively biased for both pitch trackers, which indicates a tendency towards understated pitch estimates. This tendency is more pronounced in the pitch estimates of the Multi PLL algorithm, which is also supported by the median. The standard deviation of the Single PLL is more than twice as high as the standard deviation of the Multi PLL which indicates a considerably pronounced spread around the mean value. The difference concerning the spread is clearly visible in the histograms. The slope of the Multi PLL histogram is much steeper to both sides than the slope of the Single PLL histogram. This deviating characteristic is caused by heavier oscillations of the Single PLLs pitch track which are related to the bigger frequency shift accounted for by the larger K^d value. Consequently phase errors are amplified which lead to a more sensitive and unstable tracking.

Further the histogram indicates that 77.84 % of Multi PLL pitch samples lie below the reference pitch track. It is assumed that the reason for this is a combination of the pitch trackers workings and the inharmonic nature of the instrument whose emitted

audio signal is analyzed. If an instrument has inharmonic properties, its overtones are non integer multiples of the fundamental frequency. This is particularly recognizable for stringed instruments like guitar and depends on the stiffness of actual strings [9]. The PYIN algorithm is based on the difference function which comprises the autocorrelation function [10]. Therefore, the periodicity of the whole signal, including overtones, is evaluated in order to determine the corresponding frequency. This frequency is slightly higher than the frequency of the fundamental depending on the present level of inharmonicity. The thicker the core of the string, the bigger the resulting inharmonicity [9]. Therefore, especially for the bass strings of a guitar the pitch results of the PYIN algorithm tend to be higher than the actual frequency of the fundamental. One can state that the definition of pitch varies between PLL- and autocorrelation-based pitch trackers. While the PLL-based implementations track the frequency of the fundamental, the PYIN algorithm tracks the frequency which is characterized by the periodicity of the overall signal.

The biases of the PLL-based estimators are similar and most certainly emerge from the differing pitch definition of the PYIN estimator. Therefore, the exactness of pitch estimates of PLL-based algorithms depends mainly on the spread, which is less definitive for the Multi PLL. As a result the approach described in this paper provides more precise pitch estimates than its predecessor. In order to evaluate the exactness of PYIN and Multi PLL pitch estimates the nature of pitch has to be specified more precisely and the dataset to be used needs to provide a ground truth with a finer frequency quantization.

4. CONCLUSION

The goal of this study was to develop an efficient monophonic pitch tracker, utilizing multiple PLLs, which delivers improved robustness against overtone errors and enhanced pitch track stability. The access to multiple, variably parameterized PLLs allows a much more comprehensive view of the presence, intensity and positioning of partials than a single PLL could deliver. Based on this information conclusions can be made that result in a substantial improvement of the detection rate. For the IDMT-SMT-GUITAR dataset the Multi PLL estimator has achieved a F-Measure of 82.63 %, which corresponds to an improvement of 26.26% compared to the results of its predecessor.

In future implementations especially the Recall could be improved by optimizing the individual PLL parameters and the weighting of features in order to enhance the timing and rate of the voiced detections. A problem that remains is the detection of pitch for tones with missing fundamentals. Pitch trackers which exploit the periodicity of the overall signal like autocorrelation-related approaches are capable of detecting the perceived pitch of these tones. The approach presented in this study however, requires additional logic which considers the frequency spacing of partials in order to provide this functionality. In addition to that a statistical model could be implemented to further improve the correctness and continuity of the overall pitch track. Finally, to provide real-time-capability for future implementations, the filterbank needs to be modified. An extension of the presented approach for application to polyphonic audio signals seems not practical since the configuration of the filterbank and the PLLs exploits the frequency composition of monophonic audio signals.

5. REFERENCES

[1] M. Mauch and S. Dixon, "PYIN: A fundamental frequency estimator using probabilistic threshold distributions," in *Proc. of the IEEE Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014.

[2] Edward W. Large, "Beat tracking with a nonlinear oscillator," 1995.

[3] P. Pelle and C. Estienne, "A robust pitch detector based on time envelope and individual harmonic information using phase locked loops and consensual decisions," in *Proc. of the IEEE Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014.

[4] U. Zölzer, S.V. Sankarababu, and S. Moller, "PLL-based pitch detection and tracking for audio signals," in *Proc. of the Int. Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, July 2012.

[5] U. Zölzer, "Pitch-based digital audio effects," in *Proc. of the Int. Symposium on Communications Control and Signal Processing (ISCCSP)*, May 2012.

[6] Dimitrios Giannoulis, Michael Massberg, and Joshua D Reiss, "Digital dynamic range compressor design–A tutorial and analysis," *Journal of the Audio Engineering Society*, vol. 60, no. 6, pp. 399–408, 2012.

[7] Floyd Martin Gardner, *Phaselock techniques; 2nd ed.*, Wiley, 1979.

[8] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello, "MedleyDB: A multitrack dataset for annotation-intensive MIR research," in *Proc. of the Int. Society for Music Information Retrieval Conference*, Oct. 2014.

[9] Manfred Zollner and Hochschule Regensburg, "The physics of e-guitars: Vibration, voltage, sound wave, timbre (Physik der Elektrogitarre)," Available in german language at https://hps.hs-regensburg.de/~elektrogitarre/pdfs/gesamt.pdf, accessed February 26, 2016.

[10] Alain de Cheveigné and Hideki Kawahara, "YIN, a fundamental frequency estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.