Speech Communication 116 (2020) 77–85

Contents lists available at ScienceDirect

Speech Communication

journal homepage: www.elsevier.com/locate/specom

# Robust $f_0$ extraction from monophonic signals using adaptive sub-band filtering

Pradeep Rengaswamy [a,*], M. Kiran Reddy [b], Krothapalli Sreenivasa Rao [b], Pallab Dasgupta [b]

[a] Advanced Technology Development Center,IIT Kharagpur, India
[b] Department of Computer Science and Engineering, IIT Kharagpur, India

ARTICLE INFO

*Keywords:*
Fundamental frequency
Speech
Song
Non-linear filtering
Autocorrelation
LSTM
SARGAM,

ABSTRACT

Fundamental frequency ($f_0$) extraction plays an important role in processing of monophonic signals such as speech and song. It is essential in various real-time applications such as emotion recognition, speech/singing voice discrimination and so on. Several $f_0$ extraction methods have been proposed over the years, but no one algorithm works well for both speech and song. In this paper, we propose a novel approach that can accurately estimate $f_0$ from speech as well as songs. First, voiced/unvoiced detection is performed using a novel RNN-LSTM based approach. Then, each voiced frame is decomposed into several sub-bands. From each sub-band of a voiced frame, the candidate pitch periods are identified using autocorrelation and non-linear operations. Finally, Viterbi decoding is used to form the final pitch contours. The performance of the proposed method is evaluated using popular speech (Keele, CMU-ARCTIC), and song (MIR-1K, LYRICS) databases. The evaluation results show that the proposed method performs equally well for speech and monophonic songs, and is better than the state-of-the-art methods. Further, the efficacy of proposed $f_0$ extraction method is demonstrated by developing an interactive SARGAM learning tool.

## 1. Introduction

Fundamental frequency ($f_0$) is defined as the rate of vocal folds vibrations during the production of speech or song. Although speech and song are produced by the same vocal apparatus, they differ significantly from one another in terms of production as well as perception. While speech conveys the message constrained by the language, song communicates both melody and lyrics. Further, the impact of source-filter interaction phenomena is greater in song than in speech Kadiri and Yegnanarayana (2015), Kob et al. (2011). Singers induce controlled variations in $f_0$ by rapidly changing the larynx position based on perceptual feedbacks. But, speakers in general are less concerned about the variations in $f_0$. Also, the $f_0$ range is wider and the individual sound units sustain longer in a song than in speech.

In the literature, several approaches have been proposed for $f_0$ estimation. The most widely used $f_0$ extraction methods are Praat Boersma (1993), Robust Algorithm for Pitch Tracking (RAPT) Talkin (1995), speech transformation and representation using adaptive interpolation of weighted spectrum (STRAIGHT) Kawahara et al. (2010), DIO Morise et al. (2009), summation of residual harmonics (SRH) Drugman and Alwan (2011), sawtooth waveform inspired pitch estimator (SWIPE) Lozano (2011), and YIN De Cheveign'e and Kawahara (2002). Praat and RAPT are the most popular time-domain pitch tracking approaches that estimate $f_0$ of the analysis frame by extracting local maxima of the crosscorrelation or autocorrelation function. STRAIGHT is a high-quality speech analysis and synthesis system which uses wavelet-based instantaneous frequency analysis technique for $f_0$ extraction. DIO is a pitch tracker used in the WORLD vocoder Morise et al. (2016). DIO decomposes an acoustic signal into sub-bands and estimates a novel feature called fundamentalness to identify the candidates and the final $f_0$ estimates. SRH is a frequency domain method which relies on the presence of strong harmonic peaks to estimate $f_0$. SWIPE is also a frequency domain approach which estimates $f_0$ as the fundamental frequency of the sawtooth waveform whose spectrum best matches the spectrum of the input signal. YIN is based on the difference function obtained using the autocorrelation method with a number of refinements that combine to reduce possible errors De Cheveign'e and Kawahara (2002). One of its major shortcomings is that it uses a single threshold parameter which affect the results Mauch and Dixon (2014). To address this limitation, a modified version of YIN known as probabilistic YIN (pYIN), has been proposed in Mauch and Dixon (2014). In pYIN, the threshold parameter is replaced by a parametric distribution, and several $f_0$ candidates are obtained for every frame, conditional on this prior parameter distribution. A hidden Markov model (HMM) is then employed to produce the final $f_0$ contour from the candidate $f_0$s. Thus, pYIN provides superior recall and precision while maintaining YIN's pitch accuracy.

All the above methods provide best results to analyze speech signals, but they cannot be applied to songs in a straightforward way Kob, Henrich, Herzel, Howard, Tokuda, Wolfe, 2011. This is because in speech

* Corresponding author.
*E-mail address:* rpradeep@iitkgp.ac.in (P. Rengaswamy).

https://doi.org/10.1016/j.specom.2019.11.006
Received 20 March 2019; Received in revised form 12 November 2019; Accepted 28 November 2019
Available online 29 November 2019

the $f_0$ range is narrower and source-filter coupling is relatively loose than in songs. Babacan and Drugman (2013) has shown that the speech $f_0$ trackers can be adapted to singing voice by properly tuning the set of default input parameters. In Babacan and Drugman (2013), it is observed that YIN achieved the best pitch accuracy on singing voice among the most prominent $f_0$ extraction methods. At present, YIN and its variant pYIN are widely used for estimating $f_0$ from songs. Although the $f_0$ estimation techniques can be adapted to work with speech and songs, tuning of input parameters is not a trivial task. This becomes even more difficult when an audio recording consists of both speech and song. There are several real-time scenarios where song and speech occur in the same recording. For example, *Harikatha*- a famous form of Hindu traditional discourse involves the narration of a story, intermingled with various songs relating to the story. Extracting $f_0$ in such cases is very important for applications such as query-by-humming. Since, the existing $f_0$ trackers have to be tuned separately for speech and songs, a robust speech-song discrimination algorithm is needed as a front-end to extract $f_0$ in these scenarios. Also, the existing methods fail to extract $f_0$ accurately from weakly-periodic, creaky, fricative, and transition regions Reddy and Rao (2017).

In this paper, we propose a generalized method based on adaptive sub-band filtering that can estimate $f_0$ accurately from both speech and songs. Several sub-band based $f_0$ estimation approaches have been proposed in the literature. In Ohmura (1994), a notch filter based on the lowest spectral lobe is constructed. The $f_0$ is estimated from the periodic zero crossings. The WORLD Morise et al. (2016) vocoder uses DIO Morise et al. (2009) method for $f_0$ estimation. In DIO method, initially an octave is decomposed into four channels. The variance of negative and positive going zero-crossing intervals and the interval between successive peaks and valleys are measured for each glottal cycle. The minimum variance across the bands is chosen as the valid $f_0$ candidate. Further post-processing is performed based on the contextual information to obtain a smooth $f_0$ contour. The irregular $f_0$ candidates in successive frames are treated as unvoiced frames. The sub-band based $f_0$ estimation approaches do not perform explicit voicing/unvoicing detection. In these approaches, zero-crossing measures are used in determining the $f_0$s and therefore prone to harmonic errors in frames with strong formants. In the proposed method, first the voiced regions in a speech or song are identified with a novel RNN-LSTM based voicing detection approach. Second, each voiced frame is decomposed into sub-bands. The sub-band decomposition is designed in such a way that the mono-component signal is estimated in at least one of the sub-bands. Third, the autocorrelation function is used to convert the phase-shifted, time-invariant, and periodic signal to zero-phase periodic signal. Then, non-linear techniques are employed to estimate candidate $f_0$s from the sub-bands of each frame. Finally, the optimal $f_0$ contour is derived from the candidate $f_0$s by using a Viterbi decoder. The experimental results show that the proposed method performs well for both speech and songs, without parameter tuning.

This paper is organized as follows: Section 2 elaborates the proposed $f_0$ estimation approach. Section 3 evaluates the performance of proposed and existing $f_0$ extraction methods, and provides insights into the strengths and weaknesses of these methods. Section 4 demonstrates an interactive SARGAM analyser based on the proposed $f_0$ estimator. Finally, Section 5 concludes the paper and discusses the avenues for future research.

## 2. Proposed method

The acoustic signal for speech or song is quasi-harmonic in the frequency domain and quasi-periodic in the time domain. The quasi-harmonic property is utilized to determine the number of sub-bands in the proposed method. Consider an ideal harmonic signal $S(n)$, represented by Eq. (1)

$$S(n) = \sum_{r=1}^{N} A_r \cos(2\pi r f_0 n + \phi_r) \tag{1}$$

where $rf_0$ corresponds to the $r^{th}$ harmonic of $f_0$, $rf_0 \in (0, \frac{F_s}{2})$, $A_r$ and $\phi_r$ represents the magnitude and phase of the $r^{th}$ harmonic and $F_s$ represents the sampling frequency of the acoustic signal.

Suppose, if the $f_0$ range of human voice observed in speech or song is represented by $[f_{\min}, f_{\max}]$ then $f_{\min}$ and $f_{\max}$ are the minimum and maximum frequency, respectively. The $f_0$ range of neutral speech is usually between $[50\ Hz - 400\ Hz]$ and the $f_0$ range of song is between $[50\ Hz - 800\ Hz]$. Since this work considers both speech and song, the $f_0$ range is fixed at $[50\ Hz - 800\ Hz]$. When the $f_0$ range is narrow and within an octave ($f_{\max} < 2f_0$), the finite impulse response (FIR) filtered output with passband $[f_{\min}, f_{\max}]$ oscillates at $f_0$. The mono-component signal (single frequency) refers to the signal that oscillates at $f_0$. The time interval between successive maximum or minimum peaks of the mono-component signal defines the fundamental time period ($t_0$), and subsequently, the inverse of $t_0$ represents the fundamental frequency ($f_0$).

$$f_0 = \frac{1}{t_0} \tag{2}$$

### 2.1. Decomposition of the harmonic signal

The acoustic signal is decomposed into sub-bands ($s_b(n)$) by convolving the original signal ($S(n)$) with FIR filters ($g_b(n)$).

$$s_b(n) = S(n) * g_b(n) \tag{3}$$

Let the passband of the first filter $g_1(n)$ be $[f_{\min}, 2f_{\max}]$. The bandwidth of the successive bandpass filters are narrowed down by reducing $f_{\max}$, such that at least one sub-band contains the mono-component signal. In the ideal case of a strictly harmonic signal, reducing $f_{\max}$ by an octave in consecutive filters result in mono-component signal in at least one sub-band. In practise, the causal FIR filters have a minimal transition band from passband to stopband. On the other hand, in real-time, the harmonic partials of speech or song do not lie at exact multiples of $f_0$. Due to these limitations, the bandwidth of successive filters is reduced by half an octave.

The interval between two frequencies is measured in octave scale, which is logarithmic in nature. When the ratio between two frequencies is two, then the interval between the frequencies represents one octave. Based on this measure, the frequency in Hertz scale is converted to cents scale (logarithmic scale) for determining the cut-off frequencies of the filters. The frequency to cents scale mapping is given by

$$C = 1200 \log_2 \left( \frac{f}{f_{ref}} \right) \tag{4}$$

where $f_{ref}$ represents the reference frequency (in Hz) and $C$ represents the number of cents between $f$ and $f_{ref}$. An octave is 1200 cents, so half an octave is 600 cents. The reference frequency is chosen as the minimum frequency $f_{\min}$. Thus, the cents scale equivalent of $f_{\min}$ and $f_{\max}$ is represented by $C_{\min}$ and $C_{\max}$ in Eq. (5).

$$C_{min} = 0 \tag{5}$$

$$C_{max} = 1200 \log_2 \left( \frac{f_{max}}{f_{min}} \right) \tag{5}$$

As $f_{\min}$ is the reference frequency, $C_{\min}$ is zero and $C_{\max}$ represents the number of cents in between $f_{\min}$ and $f_{\max}$. The $f_{\max}$ of successive passbands are reduced by half an octave, i.e., $C_{\max}$ is reduced by 600 cents in successive passbands, until $C_{\max}$ is non-negative. The number of sub-bands ($n_b$) is defined based on the $f_0$ range given by

$$n_b = \left\lceil \frac{C_{max}}{600} \right\rceil \tag{6}$$

The $[f_{\min}, f_{(b,max)}]$ represents the passband of the $b^{th}$ sub-band, $f_{(b,max)}$ computed as

$$f_{(b,max)} = f_{min} 2^{\left( \frac{(C_{max} - 600(b-1))}{1200} \right)}, \forall b = 1, 2, \ldots, n_b \tag{7}$$

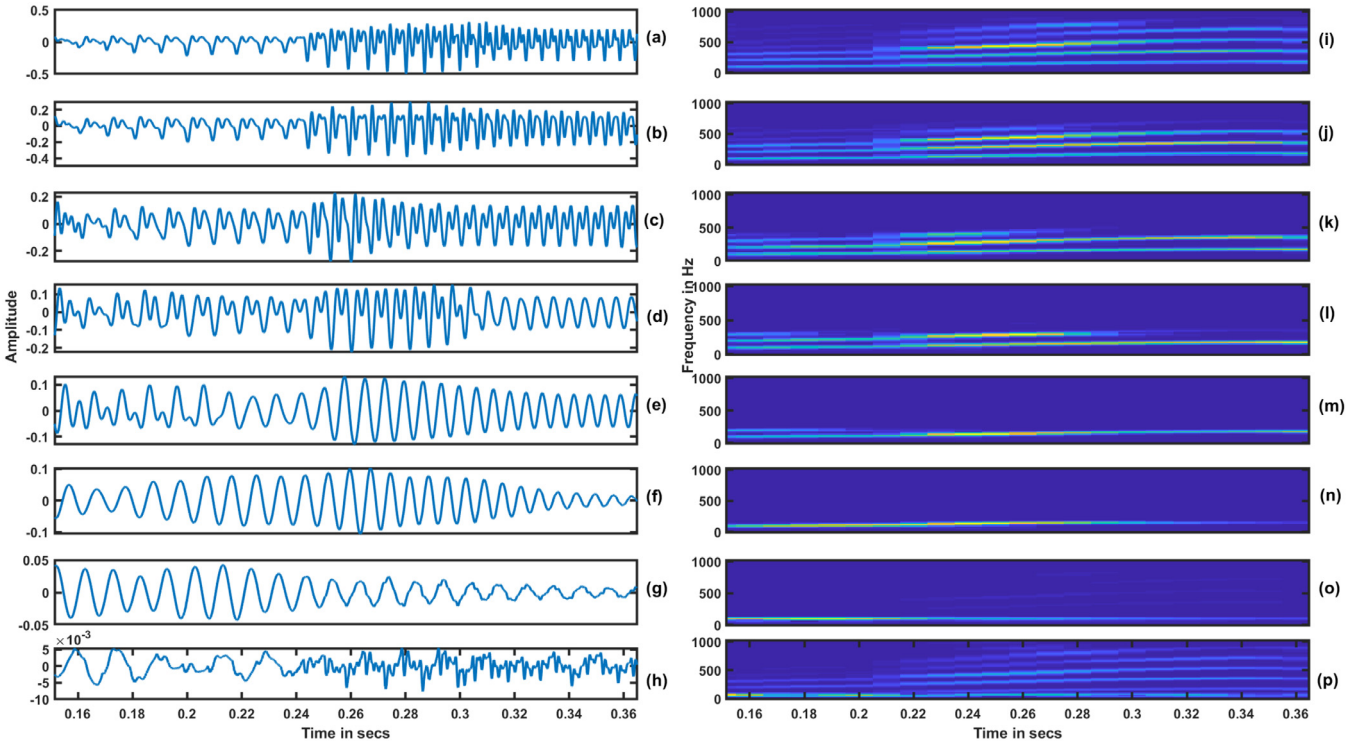


**Fig. 1.** The first column represents the time domain signals corresponding to the output of the eight sub-band filters ([50 Hz, 1600 Hz], [50 Hz, 1131 Hz], [50 Hz, 800 Hz], [50 Hz, 566 Hz], [50 Hz, 400 Hz], [50 Hz, 282 Hz], [50 Hz, 200 Hz], [50 Hz, 141 Hz]) and the second column represents the spectrograms of the corresponding time domain signals in column 1.

For e.g., when the $f_0$ range is from 50 Hz to 800 Hz, the $f_0$ range spans four octaves. The cents scale varies between 0 to 4800 cents, where $C_{\min} = 0$, $Cmax = 4800$. As the upper cut-off frequency is reduced by half an octave is successive bands, the upper cut-off frequency of sub-band $b$ is defined by $Cmax$ - $600(b-1)$. Thus the derived bandwidths of the sub-bands are [0 cents 4800 cents], [0 cents 4200 cents], [0 3600 cents], [0 cents 3000 cents], [0 cents 2400 cents], [0 cents 1800 cents], [0 cents 1200 cents] and [0 cents 600 cents]. The respective bandwidths in frequency scale are [50 Hz, 1600 Hz], [50 Hz, 1131 Hz], [50 Hz, 800 Hz], [50 Hz, 566 Hz], [50 Hz, 400 Hz], [50 Hz, 282 Hz], [50 Hz, 200 Hz], [50 Hz, 141 Hz], [50 Hz, 100 Hz], and [50 Hz, 71 Hz].

Fig. 1 shows the time and frequency domain representation of a short segment of speech signal decomposed into sub-bands using an elliptic FIR filter. In the frequency domain, the spectrum is computed for a short duration of 50 ms with 80% overlap and vertically stacked into a spectrogram. The first column in Fig. 1 represents the filtered outputs of the acoustic signal. The passband bandwidth of the FIR filters is narrowed down as we move from top to bottom. The second column represents the spectrograms of the filtered signal in column 1. The harmonic partials are de-emphasized in successive sub-bands as clearly seen from the spectrograms. Fig. 1(a) represents the time domain signal passed through $g_b(n)$ with passband [50 Hz, 1600 Hz] and the respective spectrogram is provided in Fig. 1(i). The bandwidth of the filter $g_b(n)$ is narrowed down in successive rows (as shown in Figs. 1(a) to 1(h), thereby the higher frequencies are de-emphasized in the successive sub-bands (see Figs. 1(i) to 1(p)). The time-domain signal is smoothed in successive rows as observed in Figs. 1(a)-(d) and oscillates at $f_0$ in Figs. 1(e) and (f). As the passband bandwidth of successive filters $g_b(n)$ is narrowed down, the energies of harmonic partials are suppressed as we go down from Figs. 1(i) to (p). The spectrograms shown in Figs. 1(m), (n) and (o) represent only the $f_0$, and the respective sub-bands (Figs. 1(m), (n) and (o)), represent a mono-component signal. The sub-bands with passband less than 141 Hz are not plotted since the upper cutoff frequency is less than $f_0$. The filtered output with de-emphasized $f_0$ is observed in Fig. 1(h).

### 2.2. Preprocessing

The acoustic signal is sampled at 16 kHz. Each frame is of length 50 ms with 80% overlap between successive frames. $f_0$ is estimated from the voiced frames, thus the next subsection discusses on detecting the voiced frames.

### 2.3. Voicing detection using LSTM

In the acoustic signal, the presence and absence of the glottal vibrations signify the voiced and unvoiced segments, respectively. As the voiced segments are produced by the excitation consisting of a sequence of glottal pulses, the successive voiced frames are highly correlated. The unvoiced frames include vocal tract constrictions in the absence of glottal vibrations along with silent regions. The successive unvoiced frames are less correlated. In noisy conditions, despite the distortions in some voiced frames, the information from the adjacent frames helps to learn the sequential information. In the traditional neural network architectures, each frame is independent. The voiced frames are mostly sequential, and thus the traditional neural networks are not capable of identifying the voiced frames in low excitations and transitions. In this work, we explored Long short-term memory (LSTM) a Recurrent Neural Network (RNN) architecture for capturing the sequential information. The memory in LSTMs is called cells that take as input the previous state and the current input. Internally, these cells decide in preserving or erasing the memory. The previous state, the current memory, and the input of the current frame are used in classifying a frame as voiced or unvoiced. It turns out that these types of units are very efficient at capturing long-term dependencies.

The LSTM architecture used in this work is a form of sequence-to-label classification, as shown in Fig. 2. The model is built using the

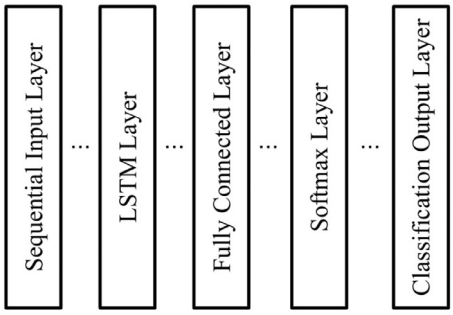


**Fig. 2.** The LSTM architecture for voiced/unvoiced frame classification.

Tensorflow framework with Keras API in Python. The input to the sequential layer is a 50 ms raw audio frame, consisting of 800-dimensional feature vector. The LSTM layer accesses the sequential frames and learns the contextual information between successive frames. The output of the LSTM layers is a sequence, which is reshaped and passed to a fully connected layer. The softmax layer squashes the output of the fully connected layer to probabilistic scores for the two classes. The Keele database is used in training because the speech signals in this database include low-voiced, creaky and breathy characteristics. The database consists of speech signals from 5 male and 5 female speakers, each of length 30 sec. The LSTM network is trained using 3 male and 3 female speakers'. A mini-batch size of 5000 is chosen to divide the training data evenly with the maximum number of epochs set to 100. The LSTM network is trained using categorical cross-entropy loss function and Adam optimizer. The default parameters for Adam optimizer used in Keras are retained during training. A callback function was defined to stop training when the validation accuracy remains unchanged for ten successive epochs. The LSTM layers consist of 186 hidden units derived based on the empirical analysis performed on the Keele database. The evaluation results for voicing detection across datasets are provided in Section 3.

### 2.4. Extraction of the mono-component signal

The voiced frames of speech or song are periodic with the fundamental time period ($t_0$).

$$R(\tau) = \frac{\sum_{i=n+1}^{n+L} S(i)S(i+\tau)}{\sum_{i=n+1}^{n+L} S(i)S(i)} \tag{8}$$

During framing, the periodic voiced segments are phase-shifted. The autocorrelation function converts the periodic signal into a zero-phase periodic signal. The $f_0$ in the autocorrelated signal remains unaltered from the actual voiced frame. The bias towards strong and weak excitation frames is removed by computing the normalized autocorrelation as in Eq. (8). The peaks in the autocorrelation sequence represent $t_0$ and its integer multiples along with peaks due to higher harmonic partials, as shown in Fig. 3(c). The higher harmonic partials are de-emphasized by decomposing the acoustic signal into sub-bands.

Fig. 3 demonstrates the significance of sub-band structure in the estimation of the mono-component signal. The voiced segment with energies near $f_0$ and higher harmonic partials is presented. The first quadrant (Fig. 3(a)) demonstrates a voiced segment with strong $f_0$ and weaker higher harmonic partials. The second quadrant (Fig. 3(b)) demonstrates a voiced segment with weaker $f_0$ and strong higher harmonic partial. The third and fourth quadrant (Figs. 3(c) and (d)) analyses the voiced segments with high energies in both $f_0$ and one harmonic partial. The $f_0$ is relatively stronger in the third quadrant, while the harmonic partial is stronger in the fourth quadrant.

In each quadrant, the first column represents the time-domain signal, the spectrum and the autocorrelation sequence computed for a voiced segment of 50 ms duration in the top, middle, and bottom figures respectively. The four figures from top to bottom in the second column represent the output of the FIR filters with the passband [50 Hz, 800 Hz], [50 Hz, 400 Hz], [50 Hz, 200 Hz], and [50 Hz, 100 Hz] respectively and the third column represents the normalized autocorrelation sequence for the sub-bands in the column 2. Four sub-bands are used in illustration, but we construct ten sub-bands for the given $f_0$ range. The spectrum of Fig. 3(a) represents a voiced frame with a strong $f_0$ and weak

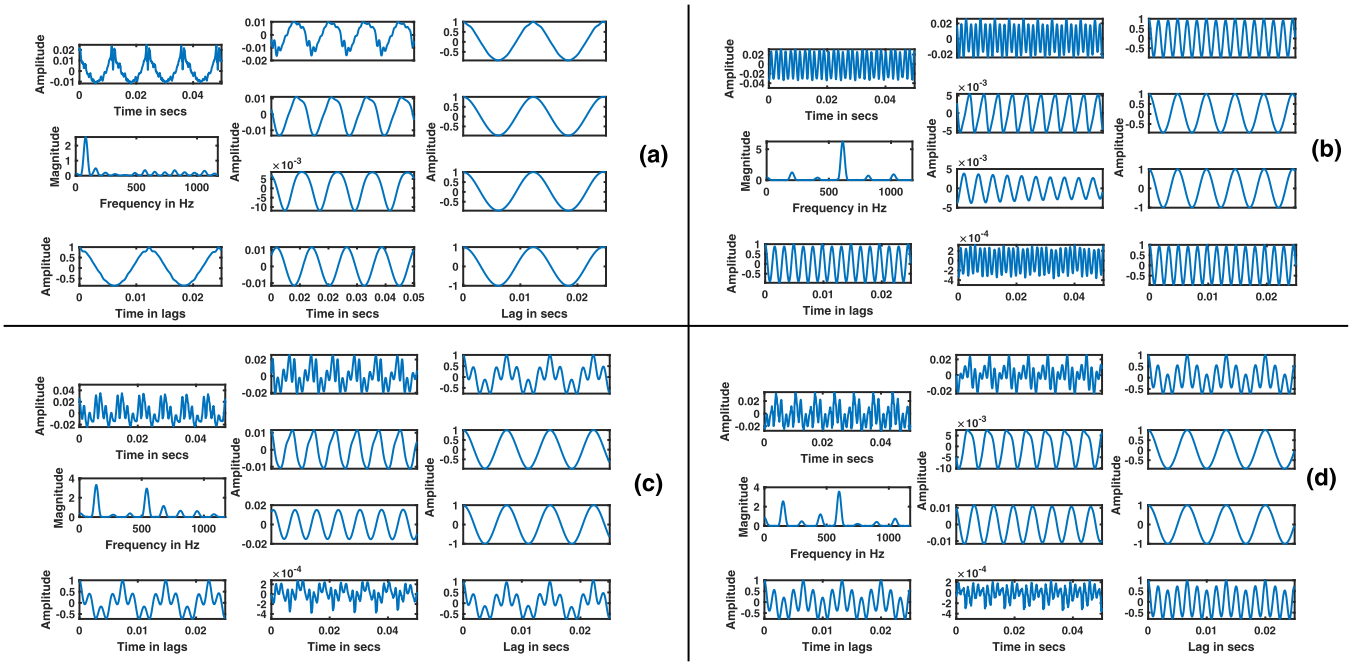


**Fig. 3.** The four quadrants represent the voiced frames chosen from four phonetic utterances. In the first column of each quadrant, the top figure represents the time-domain signal of frame *f*, the middle and the bottom figures represent the spectrum and normalized autocorrelation sequence computed for a frame *f*. The second column corresponds to the FIR filtered outputs of frame *f* passing through the passbands [50 Hz-800 Hz], [50 Hz-400 Hz], [50 Hz-200 Hz], [50 Hz-100 Hz] respectively. The third column computes the normalized autocorrelation sequence of the FIR filtered outputs shown in the second column.

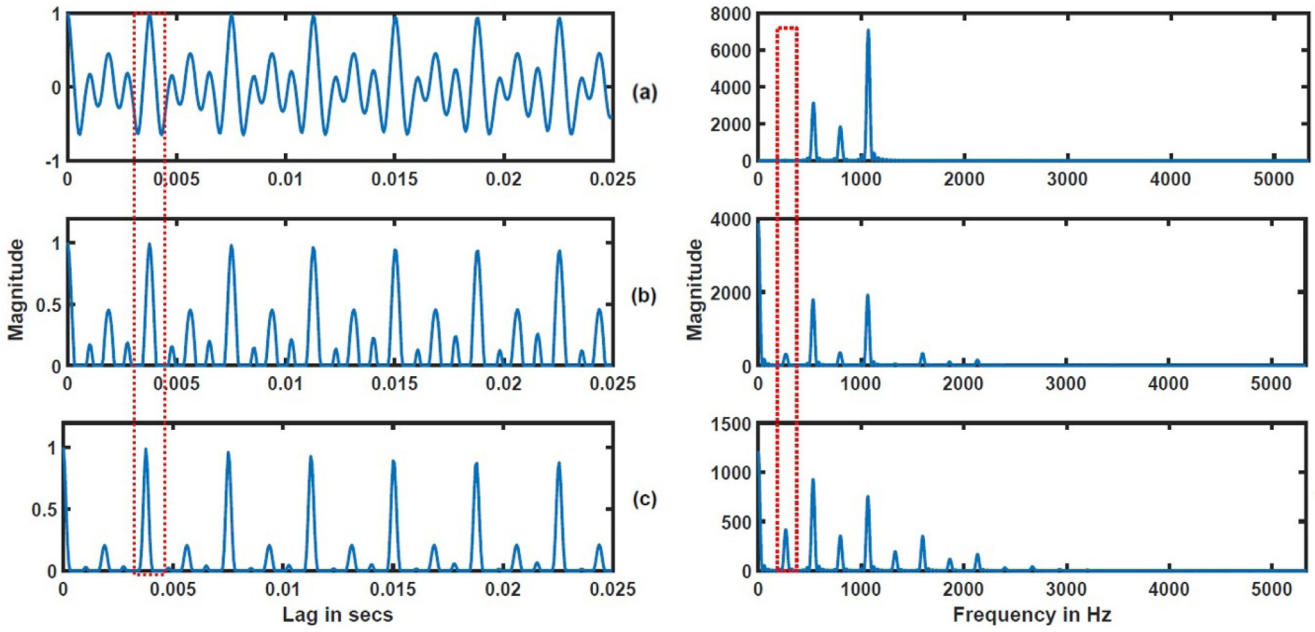


**Fig. 4.** The first column represents the computed $R_{(f,b)}(\tau)$, $R^{+}_{(f,b)}(\tau)$, and $\tilde{R}_{(f,b)}(\tau)$ and their corresponding spectra are shown in second column. The $t_0$ instants in the autocorrelation plots (shown in left column) and the corresponding $f_0$ peaks in the spectra are shown in right column with red color vertical boxes. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

higher harmonic partials. Thus, most of the sub-bands and the normalized autocorrelation outputs resemble a mono-component signal as observed in second and third column respectively. In the spectrum shown in Fig. 3(b), a strong third harmonic partial is observed with weak $f_0$. Thus the FIR filtered outputs, and the normalized autocorrelation coefficients initially modulate with time period $\frac{t_0}{3}$. In the second and third sub-band, as the higher harmonic partials are suppressed, the signal resembles a mono-component signal. The spectra of Figs. 3(c) and (d) have higher energies in both the $f_0$ and one of the higher harmonic partial. As the second and third sub-bands suppress the higher harmonic partials, both the quadrants represent a mono-component signal. As the passband is narrowed down in successive sub-bands, the FIR filtered output signal de-emphasizes $f_0$ as observed in the last sub-band of Figs. 3(c) and (d). From the examples, the proposed sub-band structure ensures that at least one sub-band represents a mono-component signal in the voiced frames.

### 2.5. Non-Linear filtering

The normalized autocorrelation sequence computed for frame $f$ in sub-band $b$ is represented by $R_{(f,b)}(\tau)$. The higher harmonic partials are suppressed by means of nonlinear filtering.

$$R^{+}_{(f,b)}(\tau) = \frac{1}{2}\left(R_{(f,b)}(\tau) + |R_{(f,b)}(\tau)|\right) \tag{9}$$

where $R^{+}_{(f,b)}(\tau)$ represents the half-wave rectified $R_{(f,b)}$ signal. Further, the coefficients of $R^{+}_{(f,b)}$ are squared to enhance the peaks in intervals of $t_0$ and suppress the spurious peaks.

$$\tilde{R}_{(f,b)}(\tau) = R^{+}_{(f,b)}(\tau) * R^{+}_{(f,b)}(\tau) \tag{10}$$

where $*$ represents the element-wise multiplication operation. As the peaks in $t_0$ intervals are prominent and closer to unity, squaring the signal has less effect on the peaks. But, the spurious peaks in-between the prominent peaks are drastically de-emphasized. The non-linear filtering process is demonstrated in Fig. 4. The left column of Fig. 4 represents $R_{(f,b)}$, $R^{+}_{(f,b)}$, and $\tilde{R}_{(f,b)}$ computed for a voiced frame, and their respective spectra are illustrated in the right column. It can be seen from the spectrum of $R_{(f,b)}$ (shown in Fig. 4 (a)) that even though the signal is periodic in $R_{(f,b)}(\tau)$, the $f_0$ is suppressed due to high energies near higher harmonic partials. By performing half-wave rectification ($R^{+}_{(f,b)}(\tau)$), the higher harmonics and $f_0$ in the spectrum are supressed and enhanced to a notable extent, respectively (see Fig. 4 (b)). After squaring $R^{+}_{(f,b)}(\tau)$, the spurious peaks are effectively suppressed and the $t_0$ intervals are further enhanced, as shown in the left plot of Fig. 4 (c). As a result, the peak corresponding to $f_0$ becomes clearly visible in the spectrum as shown in the right plot of Fig. 4 (c). This reveals that the non-linear filtering process can emphasis the $t_0$ intervals even in spectrum with weaker $f_0$ and stronger higher harmonic partials.

### 2.6. $f_0$ Estimation from the non-linear signal

In a voiced frame $f$ of band $b$, when $\tilde{R}_{(f,b)}(\tau)$ is a mono-component segment, it represents a half-wave rectified cosine signal with time period $t_0$. Thus, the properties of a cosine signal are imposed in $t_0$ estimation.

$$\tilde{R}_{(f,b)} \approx \frac{1}{2}\left(\cos(2\pi t_0 n) + |cos(2\pi t_0 n)|\right) \tag{11}$$

#### 2.6.1. Sub-band level spurious candidate elimination

The peak instants of the mono-component signal are observed at multiples of $t_0$. Due to filtering effects, smaller peaks might be present in the mono-component signal. Thus, the peaks with magnitude more than 0.5 are considered as the candidate $t_0$ and its multiples. All $t_0$ candidates should include only the consecutive multiples of the candidate $t_0$ within the set $T$. Let $T = p_1, p_2, \ldots, p_j$ be the ordered peak locations extracted from a mono-component signal. $p_1$ is a valid $t_0$ candidate, when all $p_i \in T$ can be represented as $(mp_1 \pm 0.1) \in T, \forall m \in \mathbf{Z}^{+}$. If such a condition is not satisfied, the sub-band does not contain a mono-component signal, and hence no $t_0$ is computed. The peak locations of $\tilde{R}_{(f,b)}(\tau)$ in Fig. 4 (c) are at lag values (in samples) 61, 121, 181, 241, 302 and 362 (correspond to 3.8, 7.6, 11.3, 15.1, 18.9 and 22.6 ms, respectively) for the signal sampled at 16 kHz. The lag values (in msecs) are obtained by computing the ratio between the lag values (in samples) and the sampling frequency. As all the peak locations in set $T$ can be observed to be integer multiples of 61, within the deviation of 5%, 61 is considered as the $t_0$ candidate for that sub-band corresponding to the $f_0$ value of 262.29 Hz.

#### 2.6.2. HMM Based $f_{0\ \text{tracker}}$

The magnitude of $t_0$ instants in $R_{(f)}(\tau)$ obtained from each sub-band is stored in $P_{(f,b)}$. The sub-bands with spurious candidates are assigned to zero. In total, there are eleven probable states - ten sub-bands (represent the voicing states) and one unvoiced state. The most likely sequence of states from the observations is computed using Viterbi algorithm Viterbi (1967). The state transition probabilities ($T$) are computed based on the ground truth of the Keele database. The observation probabilities $O_{(f,b)}$ are obtained from $P_{(f,b)}$. As $O_{(f,b)}$ represents the probability scores, the magnitude of each frame is normalized such that the sum of probability scores of eleven states is one. The $O_{(f,b)}$ of the unvoiced frame has unity for unvoiced state and zero for the other ten states. The initialization probability is equiprobable for all the eleven states; thus, the probability score of each state is $\frac{1}{11}$. The final $t_0$ contour is obtained based on the most likely state sequence. The librosa McFee et al. (2015) package is used in the implementation of the Viterbi decoder. For example, the magnitude and $t_0$ candidates for the ten sub-bands be {(0,0), (0,0), (0,0), (0,0), (1.034,63), (1.034,63), (0,0), (0,0), (0,0), (0,0)}. Here (0,0) indicates that no $f_0$ candidate is obtained from the sub-band. Only the fifth and sixth sub-bands represent the mono-component signal at the same location 63, the magnitude in $\tilde{R}_{(f,b)}(\tau)$ function is 1.034. The observation probabilities for the frame are {0,0,0,0,0.5,0.5,0,0,0,0,0}. Based on the initial, transition and observation probabilities the Viterbi decoder estimates the probable state in a frame. The Viterbi decoder identified state five as the probable state corresponding to the $t_0$ instant near lag 63. The respective $f_0$ value is 253.96 Hz. Thus, a dynamic programming approach is used in estimating the final $f_0$ contour. The Viterbi decoder helps in removing the octave errors and detecting $f_0$s in transition frames.

## 3. Performance evaluation

The performance of the proposed method is evaluated against six existing methods, namely, RAPT, STRAIGHT, YIN, SWIPE', DIO and pYIN. While SWIPE',RAPT, DIO and STRAIGHT are widely used for $f_0$ estimation from speech signals, YIN and pYIN are the state-of-the-art approaches for $f_0$ extraction from monophonic songs. All methods are used with their default parameter values for which they are optimized. The frame shift is fixed to 10 ms and the $f_0$ range is set to [50 Hz, 800 Hz]. In STRAIGHT method, the default window size is 80 ms and the shift between successive frames is 1 ms. When the frame shift of STRAIGHT methods was changed as per ground truth, spurious $f_0$s were encountered. So, the frame shift was retained to 1 ms and in order to match the ground truth, the output $f_0$ contour was down sampled by 10. For evaluation we use four databases, namely, Keele Plante et al. (1995), CMU-Arctic Kominek and Black (2004), MIR-1K, and LYRICS databases.

### 3.1. Databases

#### 3.1.1. CMU arctic database

This database is developed for building text-to-speech synthesis systems. The speech corpus consists of 1132 phonetically balanced sentences spoken by an American English male speaker (BDL), and an American English female speaker (SLT). The database includes the simultaneously recorded electroglottograph (EGG) signals sampled at 16 kHz. The ground truth $f_0$s are obtained by applying the RAPT algorithm on the EGG signals. This database includes mostly the modal voiced regions and some weak excitation regions.

#### 3.1.2. Keele database

The Keele database is a standard speech database for $f_0$ estimation. This database includes the speech signals and simultaneously recorded laryngograph signals of five male and five female native English speakers, each speaking a short story for a 30 s duration. The ground truth $f_0$ is available for every 25.6 ms frame with a shift of 10 ms. This database includes modal voiced, low voiced excitation, creaky and breathy voiced regions.

#### 3.1.3. MIR-1K Database

The MIR-1K database[3] is prepared by Multimedia Information Retrieval (MIR) lab. The database consists of polyphonic pop songs recorded in multiple channels. The songs (vocals) in the polyphonic music clip are available in a separate channel, which is used for analysis. The database consists of 1000 music clips with human-labeled $f_0$ values, unvoiced sounds, and vocal/non-vocal segments. This database has songs sung by many singers with ornamentations.

#### 3.1.4. LYRICS Database

The LYRICS dataset Henrich (2001); Henrich et al. (2005) has singing voices of 13 trained singers. Based on vocal type, the dataset consists of seven bass-baritones (B1 to B7), three countertenors (CT1 to CT3) and three sopranos (S1 to S3). The singing style includes ornamentations such as crescendos, arpeggios, and glissandos. The database has several low-voiced regions and dynamically varying $f_0$ contours. This database is specifically developed with vocal training consisting of ornamentations. The audio clips span for a small duration of 10 s.

### 3.2. Evaluation metrics

The standard evaluation metrics Chu and Alwan (2009) used to compare the $f_0$ extraction methods of speech signals are:

- **Voicing Decision Error (VDE):** The proportion of voiced and unvoiced frames that are misclassified.
- **Gross Pitch Error (GPE):** GPE denotes the percentage of correctly detected voiced frames in which the estimated and the reference $f_0$ differ by more than 20%.
- **Fine Pitch Error (FPE):** The standard deviation of the $f_0$ across frames that are classified as voiced and that do not come under GPE, contribute to the fine pitch error.
- **$f_0$ Frame Error (FFE):** $f_0$ frame error identifies the proportion of frames identified in VDE and GPE to the total number of frames.

The standard performance metrics[4] used to compare the $f_0$ extraction methods of monophonic music signals are as follows:

- **Voicing Detection (VD):** The proportion of voiced frames that are correctly classified.
- **Voicing False Alarm (VFA):** The proportion of unvoiced frames that are misclassified.
- **Voicing d-prime (VDP):** It identifies the inverse cumulative distribution (ICD) between the VD and VFA. The higher value indicates better discrimination between the two classes.
- **Raw Pitch Accuracy (RPA):** The proportion of the voiced frames in which the $f_0$ of the correctly classified voiced frames and misclassified unvoiced frames deviate less than $\pm\frac{1}{4}$ tone.
- **Raw Chroma Accuracy (RCA):** The proportion of the voiced frames included in RPA and the voiced frames with octave errors with the note deviation less than $\pm\frac{1}{4}$ tone are included.
- **Overall Accuracy (OA):** The proportion of correctly classified unvoiced frames and correctly classified voiced frames to the total number of frames.

The performance evaluation results obtained for speech databases are presented in Table 1. From the Table, it is evident that most of the methods fail in $f_0$ estimation of the Keele dataset. The proposed method and STRAIGHT method performs well across different datasets. The major reason behind the degradation of performance in other methods was observed due to the misclassification of whispered, breathy, voiced plosives, voiced fricatives, creaky voiced regions, weak excitation regions

[3] http://mirlab.org/dataSet/public/

[4] https://www.music-ir.org/mirex/wiki/2010:Audio_Melody_Extraction

**Table 1**
The performance of $f_0$ estimation methods against the two speech datasets (Keele dataset and CMU-Arctic dataset).

| Method | Keele Dataset | | | | CMU-Arctic Dataset | | | |
|---|---|---|---|---|---|---|---|---|
| | VDE (%) | GPE (%) | FPE (%) | FFE (%) | VDE (%) | GPE (%) | FPE (%) | FFE (%) |
| Proposed | 4.86 | 4.39 | 1.34 | 8.2 | 6.96 | 5.94 | 1.82 | 6.52 |
| pYIN | 13.56 | 8.53 | 1.69 | 14.51 | 13.17 | 9.06 | 1.96 | 12.93 |
| YIN | 15.63 | 9.62 | 1.26 | 17.74 | 13.95 | 9.42 | 1.46 | 15.36 |
| RAPT | 9.41 | 8.48 | 1.29 | 11.21 | 11.79 | 8.16 | 1.6 | 11.17 |
| STRAIGHT | 8.56 | 8.13 | 1.35 | 9.83 | 10.67 | 7.71 | 1.12 | 10.23 |
| SWIPE' | 12.25 | 1.16 | 3.25 | 12.73 | 11.66 | 0.46 | 2.4 | 11.87 |
| DIO | 8.77 | 1.81 | 1.91 | 9.59 | 8.07 | 2.24 | 1.55 | 9.21 |

**Table 2**
The performance of $f_0$ estimation methods against the two Song datasets (MIR-1K dataset and LYRICS dataset).

| Method | MIR-1K Dataset | | | | | | LYRICS Dataset | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | VD | VFA | VDP | RPA | RCA | OA | VD | VFA | VDP | RPA | RCA | OA |
| Proposed | 0.99 | 0.14 | 3.85 | 0.97 | 0.97 | 0.94 | 0.99 | 0.12 | 3.26 | 0.96 | 0.96 | 0.95 |
| pYIN | 0.99 | 0.2 | 4.06 | 0.96 | 0.96 | 0.92 | 0.98 | 0.24 | 4.17 | 0.93 | 0.93 | 0.92 |
| YIN | 0.96 | 0.37 | 2.21 | 0.86 | 0.86 | 0.78 | 0.99 | 0.41 | 3.58 | 0.91 | 0.92 | 0.89 |
| RAPT | 0.99 | 0.36 | 2.85 | 0.91 | 0.92 | 0.83 | 0.99 | 0.32 | 3.7 | 0.90 | 0.91 | 0.87 |
| STRAIGHT | 0.98 | 0.23 | 2.36 | 0.92 | 0.92 | 0.89 | 0.97 | 0.26 | 2.84 | 0.87 | 0.89 | 0.85 |
| SWIPE' | 0.99 | 0.07 | 3.96 | 0.9 | 0.91 | 0.91 | 0.98 | 0.34 | 3.18 | 0.9 | 0.9 | 0.85 |
| DIO | 0.96 | 0.11 | 3.17 | 0.92 | 0.92 | 0.91 | 0.97 | 0.21 | 4.05 | 0.88 | 0.88 | 0.88 |

as unvoiced segments. This is were the proposed method trains a sequence aware RNN-LSTM network for voiced/unvoiced classification. The voicing decision is dependent on the input representation and the contextual information. The VDE based on the RNN-LSTM model outperforms the signal processing approaches in voicing decisions. During $f_0$ estimation, almost all methods perform well in modal voiced regions. The DIO and the proposed method detect $f_0$ in most of the voiced regions. The STRAIGHT method fails in creaky and low-voiced regions. The lower the VDE, the method is capable of detecting the whispered, breathy, voiced plosives, voiced fricatives and creaky voiced regions. As Keele database includes more non-modal voiced regions, the FFE is higher. The VDE is high for SWIPE while the GPE is lower; this shows that most of the detected voiced frames are from the modal voiced regions. The non-modal voiced frames are quasi-periodic and $f_0$ estimation deviates in specific frames by more than 20%. Thus, VDE is inversely proportional to GPE in most of the methods. The FPE computed over the correctly classified voiced frames is relatively lower across all the methods.

The parameters of the existing speech-based $f_0$ estimation approaches are tuned when applied over songs as suggested by Babacan Babacan and Drugman (2013). Table 2 shows the performance comparison of the proposed method with state-of-the-art methods for song databases. The DIO, SWIPE', RAPT and STRAIGHT methods fall into octave errors, due to the presence of high-energy harmonics in case of monophonic songs. Hence, they exhibit lower RPA and RCA for both the song databases. The pYIN performs well but fails to detect the transitions that occur for a larger duration between notes. The performance of proposed method is better than other approaches for songs. The RNN-LSTM for voicing detection and signal processing approaches for $f_0$ estimation have greatly supported for higher performance with proposed approach. The overall results show that the proposed $f_0$ extraction method is more robust to variations across speech and songs compared to other methods.

Informally, we also evaluated the performance of $f_0$ extractors on Indian Institute of Technology, Kharagpur Simulated Emotion Hindi Speech Corpus (IITKGP-SEHSC) Koolagudi et al. (2011). This corpus is recorded from the professional artists in Gyanavani FM radio station to produce various emotions. It includes neutral, angry, disgust, fear, happy, sad, sarcastic, and surprise emotions. For analysis, we considered 10 speech files (each of duration 3 seconds) from each emotion. It is observed that the performance of proposed method is better than the existing methods for all the emotions. Fig. 5 shows the $f_0$ contours estimated using the state-of-the-art and proposed methods for neutral speech, monophonic songs, and emotional speech (surprise). From the figure, it is evident that, unlike existing methods, the proposed method can extract $f_0$ accurately in all cases without parameter tuning.

Further, we analyzed the computational complexities of the proposed and existing approaches. The source codes of pYIN and YIN are obtained from ESSENTIA music library Bogdanov et al. (2013) available in Python. The source codes of SWIPE[5], STRAIGHT[6], DIO[7], and RAPT[8] are obtained from reliable sources in Matlab. Five audio recordings of around 30 s durations are used for computing the time complexity. An Intel i7-4790 CPU with 8 GB RAM system is used to perform all the analysis. The average times required to estimate $f_0$ from 30 s audio clips are 1.29, 1.89, 3.37, 6.02, 7.15, 9.92, 12.39 s for the DIO, YIN, pYIN, SWIPE, RAPT, STRAIGHT and the proposed method, respectively. The source code of the proposed method is developed in Python. The code is not optimized for parallel processing. Each sub-band is processed sequentially and thus requires more time for execution. The major time is spent in estimating signal processing features, while the LSTM model takes less than a second for computation.

## 4. Automated SARGAM Learning System (ASLS)

Most of the MIR related tasks require accurate estimation of the fundamental frequency ($f_0$). In this section, we employ the proposed $f_0$ estimation method to develop an audio-visual SARGAM learning system for Indian Classical Music. The system aims to detect the wrongly rendered notes by amateur singers. In Hindustani and Carnatic music, the vocal training practices include rendering notes called SARGAM, which is equivalent to arpeggio in Western music. Indian art music considers twelve semitones (notes) in an octave. The detailed explanation of Hindustani music and note intervals is available in Rao and Rao (2014). The

[5] https://github.com/kylebgorman/swipe
[6] https://github.com/shuaijiang/STRAIGHT
[7] https://github.com/mmorise/World
[8] http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html

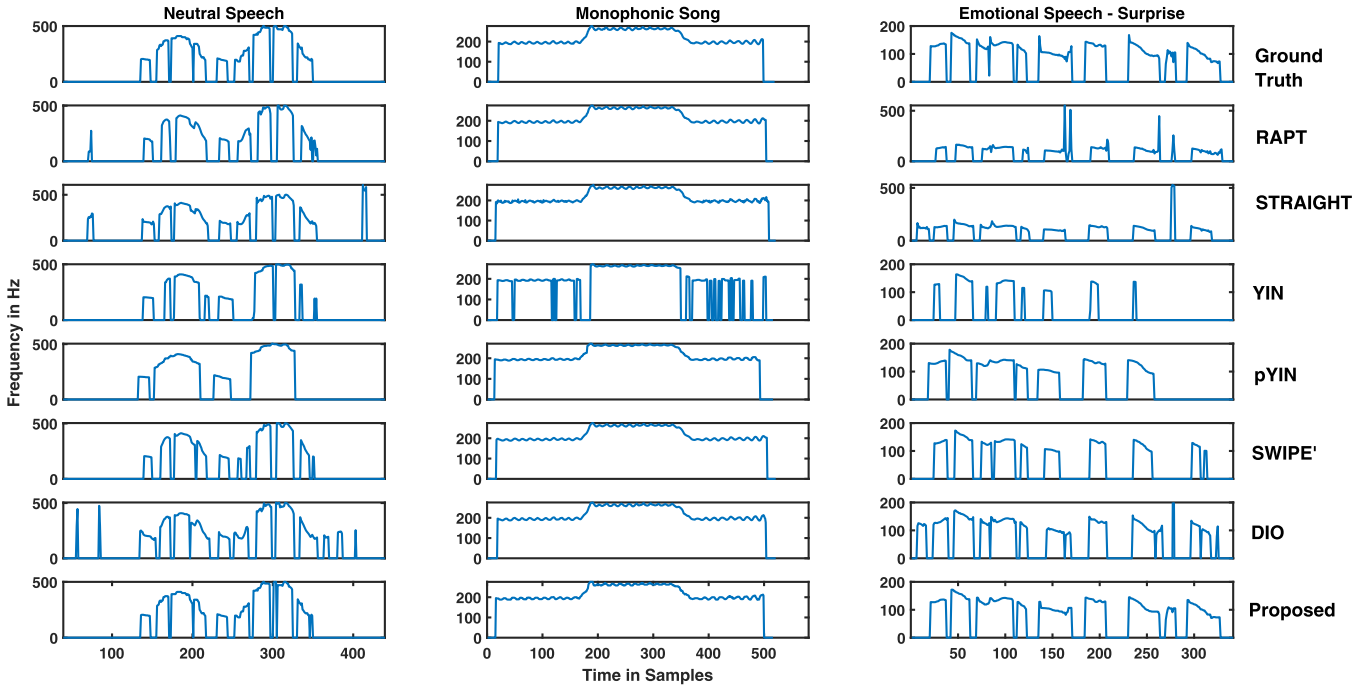


**Fig. 5.** Illustration of the $f_0$ contour over a sample neutral speech, monophonic song and emotional speech is represented in successive columns with reference Ground truth, RAPT, STRAIGHT, YIN, pYIN, SWIPE', DIO and Proposed $f_0$ estimation are represented in successive rows respectively.

main objective of the SARGAM practice is to perceive the perceptual difference between notes and replay the notes as rendered by the expert singer. The singer is free to choose his base note or tonic in Indian Classical Music. The base note defines the notes, octaves and suitable $f_0$ range of the singer. The $f_0$ range of male, female and children are different. Usually, the female singer prefers the base note between 160–230 Hz while a male singer prefers his base note between 100–180 Hz and children choose base note between 280–400 Hz.

During the initial SARGAM learning sessions, the learner is less aware of the perceptual deviations. Most of the cases, this ceases the interest in practicing singing in the initial stages of learning. The learner is concerned in uttering the lyrics of the note rather than the actual song. The visual representation of the $f_0$ contour provides preliminary clues on the relation of $f_0$ with singing. The technical challenges in developing an automated system include tonic identification of expert and amateur singers since SARGAM need not start with the base note. The tempo and rhythm of the singers vary, but as SARGAM is a sequence of notes sung in succession, each voiced segment is considered a note. The query-by-humming methods Kotsifakos et al. (2012) that compared two renditions without tonic perform $f_0$ normalization using mean subtraction. Since the amateur singer is unaware of singing, the $f_0$ normalization can be biased towards wrongly sung notes.

The learner clips may include characteristics of both the neutral speech and monophonic songs. Thus the $f_0$ estimation approach chosen should work reasonably well for both speech and songs. The tonic of the amateur singer is not known. This motivated towards developing a tonic independent automated SARGAM practicing application with audio-visual feedbacks. A sinusoidal synthesizer Bogdanov et al. (2013) is developed that masks the lyrics and synthesizes $f_0$. The visual system provides the deviation of $f_0$ contour (in cents) of the learner from the expert singer. Thus, the synchronized perceptual and visual feedback provide the learner to self-realize the errors.

SARGAM rendition consists of a sequence of notes played in succession. Each continuous, voiced segment is considered as a note. The $f_0$ contour of a note is relatively flat. Thus, the median $f_0$ is computed to represent a note. When $k$ notes are sung by the expert singer (E) and repeated by the amateur singer (L), the singer L is in-tune with E, when

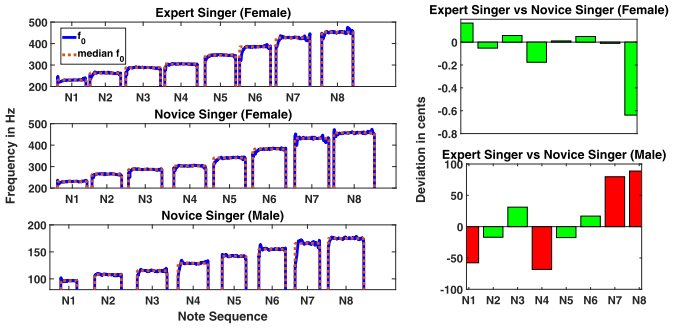


**Fig. 6.** The first column illustrates the note sequence sung by the expert singer, amateur female, and male singer, respectively. The bar plot in the second column compares the deviation of sung notes from expert singer to amateur female and male singer, respectively. The bar plot projects the wrongly sung notes in red color (i.e., the notes with deviation more than 50 cents). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the number of cents between E and L are constant across notes. Let $c_j$ represent the number of cents between the median $f_0$ of expert and amateur singer as computed in Eq. (12)

$$\left|1200\log_2\left(\frac{E_j}{L_j}\right)\right| = c_j, \forall j \in 1,2,\ldots,k \tag{12}$$

where $E_j$ and $L_j$ denotes the $j^{th}$ note sung by the expert and amateur singer, respectively, When all the notes are exactly in-tune, $c_j$ are same or $dev_j$ is zero. When notes are out-of-tune, $c_j$ vary between successive notes, thus $dev_j$ is greater than zero.

$$dev_j = |c_j - median(c)| \tag{13}$$

In Carnatic or Hindustani music the interval between successive notes is 100 cents. Thus, a note is considered as in-tune if the deviation between two singers is less than 50 cents. Fig. 6 represents the $f_0$ contour extracted based on the proposed method for both the expert

**Table 3**
Preference scores obtained by comparing the proposed method with two existing methods.

| Method | Proposed vs pYIN | Proposed vs RAPT |
|---|---|---|
| Proposed Method preferred (%) | 34.47 | 80.5 |
| Equivalent (%) | 57.5 | 17.5 |
| Other Method preferred (%) | 8.03 | 2 |

(female) and the two amateur singers (female and male respectively). The amateur female singer renders the sequence of notes as sung by the expert singer. The deviation measure for the eight notes is represented using a bar plot. As both expert and amateur singer (female) are in tune, the deviation is very minimal. But in case of the amateur singer (male), four notes deviate more than 50 cents, represented using red color in the bar plot. This provides the visual interpretation of wrongly sung notes.

Further, the sine model synthesizer is developed based on the $f_0$ contour estimated using the proposed method. This masks the lyrics and focuses on the perceptual $f_0$ contour variations. The learner is advised to listen to the humming of the expert and amateur singer for perceptual interpretation. The note level deviations are provided as a bar chart for visual interpretation. The perceptual test is performed over five expert SARGAM clips and three male and female amateur singers. Five arpeggio clips from LYRICS vocal training dataset are also added to the perceptual evaluation. Most of the existing monophonic song based $f_0$ estimation prefer pYIN and RAPT methods. The $f_0$ contour is estimated from pYIN, RAPT, and the proposed method and synthesized using a sine model synthesizer. All the $f_0$ contours are smoothed with a fifth-order median filter. For the subjective evaluation, the synthesized $f_0$ contour based on RAPT, pYIN, and the proposed method of expert and amateur singers are provided to ten listeners in random order. Three out of the ten listeners have a background in singing. All listeners were initially exposed to the actual rendition so that the perceptual mapping can be performed intuitively. The performance scores obtained by comparing the three methods are shown in Table 3. The ranking preference of the listeners is proposed, pYIN, and then the RAPT method. For some clips both the proposed and pYIN methods were preferred. In some SARGAM renditions, the amateur singers utter breathy voiced notes at lower $f_0$s, and some notes resemble the characteristics of the neutral speech. Thus, some breathy voiced notes with lower amplitude are misclassified as unvoiced region by both pYIN and the RAPT method. The RAPT method is prone to octave errors.

## 5. Conclusions

There exist separate approaches for accurate $f_0$ estimation from speech and monophonic songs. But no single algorithm performs equally well for both speech and song. In this paper, we proposed an $f_0$ estimation method, which estimates $f_0$ accurately from both speech and monophonic songs. In the proposed method, the input signal (speech or song) is decomposed into sub-bands based on the harmonic property of the human voice. This decomposition ensures that at least one sub-band represents a mono-component signal. As $f_0$ estimation is performed on the voiced frames, an RNN–LSTM model is developed for detecting voiced frames. Each sub-band is phase normalized using autocorrelation function, and non-linear signal processing approaches to emphasize $f_0$. The Viterbi decoder estimates the final $f_0$ contour from the candidates obtained from the sub-bands. The performance evaluation results show that the proposed method performs well for both speech, song and emotional speech. In future, the efficacy of the proposed method may be thoroughly analyzed for emotional speech and infant cry. Further, the robustness of the proposed method can be studied under noise and reverberant conditions. Also, the proposed method can be used to extract $f_0$ for various applications like speech synthesis.

## Declaration of Competing Interest

The authors declare that they do not have any financial or nonfinancial conflict of interests.

## Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.specom.2019.11.006.

## References


Babacan, O., Drugman, T., 2013. A comparative study of pitch extraction algorithms on a large variety of singing sounds. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 7815–7819.

Boersma, P., 1993. Accurate short-term analysis of fundamental frequency and the harmonics-to-noise ratio of a sampled sound. In: Proceedings of the Institute of Phonetic Sciences, Vol. 17, pp. 97–110.

Bogdanov, D., Wack, N., Gómez, E.G., Gutiérrez, E., Gulati, S., Boyer, P.H., Mayor, O., Trepat, G.R., Salamon, J., González, J.R.Z., Serra, X., 2013. Essentia: An audio analysis library for music information retrieval. In: 14th Conference of the International Society for Music Information Retrieval (ISMIR), pp. 493–498.

Chu, W., Alwan, A., 2009. Reducing f0 frame error of f0 tracking algorithms under noisy conditions with an unvoiced/voiced classification frontend. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Vol. 1, pp. 3969–3972.

De Cheveign'e, A., Kawahara, H., 2002. YIN, A fundamental frequency estimator for speech and music. J. Acoust. Soc. Am. 111 (4), 1917–1930.

Drugman, T., Alwan, A., 2011. Joint robust voicing detection and pitch estimation based on residual harmonics. In: Proceedings of Interspeech, pp. 1973–1976.

Henrich, N., 2001. Study of the glottal source in speech and singing: modeling and estimation, acoustic and electroglottographic measurements, perception. Theses, Université Pierre et Marie Curie-Paris VI.

Henrich, N., d'lessandro, C., Doval, B., Castellengo, M., 2005. Glottal open quotient in singing: measurements and correlation with laryngeal mechanisms, vocal intensity, and fundamental frequency. J. Acoust. Soc. Am. 117 (3), 1417–1430.

Kadiri, S.R., Yegnanarayana, B., 2015. Analysis of singing voice for epoch extraction using zero frequency filtering method. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 4260–4264.

Kawahara, H., Estill, J., Fujimura, O., 2010. Aperiodicity extraction and control using mixed mode excitation and group delay manipulation for a high quality speech analysis, modification and synthesis system STRAIGHT. Second International Workshop on Models and Analysis of Vocal Emissions for Biomedical Applications.

Kob, M., Henrich, N., Herzel, H., Howard, D., Tokuda, I., Wolfe, J., 2011. Analysing and understanding the singing voice: recent progress and open questions. Current Bioinform. 6 (3), 362–374.

Kominek, J., Black, A., 2004. The CMU arctic speech databases. In: Proceedings of 5th International Speech Communication Association. Speech Synthesis Workshop. Pittsburgh, PA, USA, pp. 223–224.

Koolagudi, G.S., Reddy, R., Yadav, J., Rao, K.S., 2011. IITKGP-SEHSC: Hindi speech corpus for emotion analysis. In: IEEE International conference on devices and communications (ICDeCom), pp. 1–5.

Kotsifakos, A., Papapetrou, P., Hollmén, J., Gunopulos, G., Dimitrios, A.V., 2012. A survey of query-by-humming similarity methods. In: Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments, ACM, p. 5.

Lozano, A. C., 2011. Swipe: A sawtooth waveform inspired pitch estimator for speech and music.

Mauch, M., Dixon, S., 2014. pYIN: A fundamental frequency estimator using probabilistic threshold distributions. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 659–663.

McFee, B., Raffel, C., Liang, D., Dawen, E., Daniel, P., McVicar, M., Battenberg, E., Nieto, O., 2015. librosa: Audio and music signal analysis in python. In: Proceedings of the 14th python in science conference, Vol. 8.

Morise, M., Kawahara, H., Katayose, H., 2009. Fast and reliable f0 estimation method based on the period extraction of vocal fold vibration of singing voice and speech. Audio Engineering Society Conference: 35th International Conference: Audio for Games.

Morise, M., Yokomori, F., Ozawa, K., 2016. WORLD: A vocoder-based high-quality speech synthesis system for real-time applications. IEICE Trans. Inf. Syst. 99 (7), 1877–1884.

Ohmura, H., 1994. Fine pitch contour extraction by voice fundamental wave filtering method. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2. Pp. II–189

Plante, F., Meyer, G.F., Aubsworth, W.A., 1995. A pitch extraction reference database. In: Proceedings of Eurospeech. Madrid, Spain, pp. 837–840.

Rao, S., Rao, P., 2014. An overview of hindustani music in the context of computational musicology. In: Taylor & Francis Journal of new music research, Vol. 43, pp. 24–33.

Reddy, M.K., Rao, K.S., 2017. Robust pitch extraction method for the HMM-based speech synthesis system. IEEE Signal Process. Lett. 24 (8), 1133–1137.

Talkin, D., 1995. A robust algorithm for pitch tracking (RAPT). Speech Coding Synth. 495, 518.

Viterbi, A., 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. In: IEEE transactions on Information Theory, Vol. 13, pp. 260–269.