Statistical Regression Models for Noise Robust F0 Estimation Using Recurrent Deep Neural Networks

Akihiro Kato *Member, IEEE,* and Tomi H. Kinnunen, *Member, IEEE*

***Abstract*—The fundamental frequency (F0) in a speech signal, which corresponds to pitch, is one of the key features involved in a variety of speech processing tasks. Therefore, accurate F0 estimation has remained an important problem to be solved over decades. However, this problem is difficult, especially in low signal-to-noise ratio (SNR) conditions with unknown noise. In this work, we propose new approaches to noise-robust F0 estimation using recurrent neural networks (RNNs). Recent F0 estimation studies exploit deep neural networks (DNNs), including RNNs, to classify acoustic features into quantized frequency states. In contrast to these classification approaches, we put forward a *regression* method for F0 tracking, which is accomplished with RNNs. To this end, we propose two variants. Our first model predicts the (scalar) F0 value directly from a spectrum, while our second model predicts a target sinusoidal waveform (with the desired F0) from the raw speech waveform.**

Our experiments with the *pitch tracking database from Graz University of Technology* (PTDB-TUG), contaminated by additive noise (NOISEX-92), demonstrate the improvement of the proposed approaches in terms of the gross pitch error (GPE) and fine pitch error (FPE) rates by more than 35 % at SNRs between -10 dB and +10 dB against a well-known, noise-robust F0 tracker, PEFAC. Furthermore, our methods outperform state-of-the-art neural network-based approaches by more than 15 % in terms of both the FPE and GPE rates over the abovementioned SNR range.

***Index Terms*—fundamental frequency, F0, pitch, waveform-to-sinusoid regression, regression model, recurrent neural networks**

I. Introduction

THE *fundamental frequency* (F0) is the lowest frequency of a periodic signal. In the human speech production process, motion of the vocal chords generates a glottal pulse, which then resonates at the vocal tract cavities to generate voiced speech. The pitch period (the reciprocal of F0) determines the harmonic structure of voiced speech, while the pitch variation over different speech sounds leads to speech prosody. F0 estimation is an essential technique for speech analysis and generation, including speech synthesis [1], voice conversion [2], prosody analysis [3], speech coding [4], model-based speech enhancement [5], [6] and speaker and language identification [7], [8]. Methods or devices that estimate F0 from short-time segments of speech are known as *F0 trackers*.

Studies in speech signal processing have proposed a variety of F0 trackers over the past decades for F0 estimation. In particular, autocorrelation-based approaches in the time domain, such as the *robust algorithm for pitch tracking* (RAPT) [9], *YIN* [10] and *probabilistic YIN (pYIN)* [11], are usually sufficiently accurate in clean conditions. However, their accuracy drops substantially in noisy conditions [12], [13], which has motivated further research into noise-robust F0 trackers. For example, *pitch estimation filter with amplitude compression* (PEFAC) [14] is an example of a successful noise-robust F0 tracker. PEFAC applies matched filters and autocorrelation in the log-frequency domain to achieve noise robustness. Nevertheless, the accuracy remains unsatisfactory in severe noise conditions, such as signal-to-noise ratios (SNRs) less than or equal to 0 dB [13], [15].

Different approaches using *machine learning* have also been studied in addition to the preceding signal processing approaches to improve noise robustness. *Generative models* with *Gaussian mixture models* (GMMs) and *hidden Markov models* (HMMs) to model F0 contours are representative of these machine learning approaches [16], [17]. Similar models have been used not only for F0 estimation from noisy speech but also for F0 generation in text-to-speech (TTS) applications [18]. In the latter case, which is not considered in this study, F0 is generated from linguistic features rather than being estimated from an acoustic waveform.

More recently, different variants of *deep neural networks* (DNNs) [19], such as *convolutional neural networks* (CNNs) and *recurrent neural networks* (RNNs), have demonstrated substantial improvements over classic generative models in many speech classification tasks. These *discriminative* models produce posterior probabilities for each state (class) for given inputs with highly complex nonlinear mapping and can be particularly suitable for modeling high-dimensional, correlated data such as raw speech waveforms or spectrograms. For F0 processing tasks, the latest research exploits DNNs [20]–[22], CNNs [13] and RNNs [20], [23] for both F0 tracking and generation. These methods classify the input acoustic or linguistic features into *quantized frequency states* (similar to the 12-semitone representation of musical notes within an octave). Such a *classification* approach to F0 tracking allows the 'plug-in' of any classifier and can be motivated from the insensitivity of the human auditory system to small pitch changes: by using enough F0 states, continuous variations in F0 may be replaced with a quantized version without a noticeable difference to average listeners.

The classification approach to F0 tracking, however, is not without problems. Because of the quantized presentation, the F0 contour becomes less natural [15], [24]. There is a trade-off between the tracking accuracy and naturalness of F0: while a larger number of quantization levels leads to a closer approximation to a continuous F0 contour, classification errors (*i.e.*, prediction of the wrong F0 state) are also more likely to occur. Furthermore, not all speech applications are targeted for human listeners. Speaker and language characterization, microprosody analysis, and spoofing attack detection (to distinguish human speech from computer-generated speech) are

TABLE I
SELECTED BASELINE F0 TRACKERS AND THE PROPOSED APPROACHES. THE BOLD FONT DENOTES OUR PROPOSAL. THE PRE- AND POST-PROCESSING OPERATIONS INCLUDE, FOR EXAMPLE, PRE-EMPHASIS FILTERING, MATCHED FILTERING AND DYNAMIC PROGRAMMING.

Base Technique	Method	Key Feature	Observation	Pre-/Post-Processing	Offline Training	Unvoiced Class
Signal Processing	RAPT [9] YIN [10] pYIN [11]	Autocorrelation	Waveform in the time domain	Required	Not Required	Continuous
	PEFAC [14]		Spectrum in the log-frequency domain			
Machine Learning	Classification	DNN / CNN / RNN Classification	Spectrum in the log-frequency domain [20] Waveform in the time domain [13], [22]	**Not Required**	**Required**	Isolated
	Regression	**RNN Regression**	**Spectrum in the frequency domain** **Waveform in the time domain**			**Continuous**

applications that may benefit from a fine-grained F0 representation. These observations suggest that machine learning-based F0 tracking is more naturally cast as a *regression* task rather than a classification task. Prior literature on neural-network-based regression approaches to F0 tracking is surprisingly scarce, although there are related models in F0 prediction for TTS tasks, *e.g.*, *Deep Voice 2* [25]. One reason why classification approaches have received more attention could be related to another important subproblem on F0 estimation: *voicing decision*. In classification approaches, voicing information can be represented by augmenting an additional state to represent the unvoiced class, whereas regression approaches must encode the voicing information differently.

This work is an extension of our two recent, preliminary studies [15], [24], in which we introduced two types of RNN regression models for F0 estimation. Our first approach [15] uses the spectral magnitude to predict the (scalar) F0 value directly, while our second model [24] first maps the raw waveform input into a sinusoid (vector) oscillating with F0. F0 can easily be extracted from this sinusoid using standard signal processing operations (here, through autocorrelation). Importantly, neither of our approaches requires additional pre-processing or post-processing operations, such as pre-emphasis filtering, matched filtering or dynamic programming, which are essential components of the classic signal processing-based approaches [9], [10], [14]. Nonetheless, as a supervised, data-driven approach, we do require a training corpus with ground-truth F0 values to train the statistical models. The key features of the selected F0 tracking approaches are summarized in Table I.

A summary of the novel contributions with respect to our preliminary studies [15], [24] is as follows:

- A comparison of our two different methods under unified evaluation conditions, including new analyses of certain experimental parameters (*e.g.*, the subsequence length in the first method)
- A detailed analysis for the inescapable trade-off between quantization and classification errors in the existing classification approaches (Section II)
- A novel, principled *bi-Gaussian* voicing detector integrated into our F0 trackers.

We substantially extend the background and preliminary comparisons of [15] and [24] towards a self-contained representation. First, in this study, both methods are compared against each other and against a number of reference F0 trackers, including signal processing and machine learning approaches. Second, to motivate the need for regression approaches, Section II provides insight into the inescapable trade-off between quantization and classification errors in the classification approach. To the best of our knowledge, such an analysis has not been presented in any prior work. Finally, neither [15] nor [24] addressed *voicing detection* in a principled way (in [24], we used an *ad hoc* threshold approach, where a heuristically determined, single voicing threshold was applied to all speakers and utterances). Due to the varied pitch range of speakers, it is generally challenging to use a common threshold for everyone. Therefore, one key contribution of this work, inspired by similar approaches used in speech synthesis [26] and speech activity detection [27], [28], is to equip our methods with *adaptive* voicing detectors. Our voicing detector is based on bi-Gaussian modeling of the outputs provided by the regression models, trained in an *unsupervised* manner for each recording. Therefore, it requires neither offline training data nor training labels. As an overall summary, our proposed approach provides F0 estimation with improvements in terms of both noise robustness and naturalness. Therefore, it can contribute to the further development of various speech applications because obtaining accurate F0 contours is vital for many of the speech applications mentioned above.

II. CLASSIFICATION APPROACH TO F0 ESTIMATION

DNN-based classification is a predominant approach to F0 tracking [20]. Despite showing improvements over rule-based methods, it has a number of shortcomings. Frequency quantization is one of the main obstacles in obtaining natural F0 contours [24]. To suppress the influence of quantization, one may use narrower quantization intervals, though potentially at the expense of decreased classification accuracy. With the motivation for the need for regression approaches (detailed in Section III), in this section, we elaborate on this trade-off using both theoretical and empirical analyses.

A. Classification Approach

As illustrated in Fig. 1, classification approaches to F0 estimation [13], [20]–[22] utilize a discriminative model to derive the posterior probabilities of each frequency state given input features, $P(s_k|\mathbf{x}_i)$. Here, s_k denotes the k-th frequency state ($k = 1, 2, \ldots, K-1$), and $\mathbf{x}_i$ is the acoustic feature vector of the i-th frame ($i = 0, 1, \ldots, I-1$). Given $P(s_k|\mathbf{x}_i)$ and the transition probability, γ_{kj}, which represents the transition probability from the previous frame state, s_j, to s_k, one finds the most likely frequency state, $s_{y'}$, by means of the Viterbi algorithm [29], [30]. The quantized frequency associated with $s_{y'}$ represents the estimate of F0 at the i-th frame, $\hat{f0}_i$.

$$\hat{f0}_i = C(s_{y'}) \tag{1}$$
$$y' = \arg\max_k P(s_k|\mathbf{x}_i). \tag{2}$$

Here, $C(\cdot)$ denotes a mapping function to associate s_k with a quantized frequency value, $f^k \in \{f^1, \ldots, f^{K-1}\}$. f^k is usually represented by the center point of each frequency interval based on uniform division.

An F0 tracker is complete only when equipped with a *voicing detector*: a binary classifier to predict the voiced/unvoiced state of a speech frame. In the classification approach, voicing detection is straightforward: one adds an extra unvoiced state, s_0, representing a discontinuous space to the output layer. This approach requires no modifications to the training procedures, and at the runtime of the F0 estimation stage, the problems of F0 estimation (read from states $s_1, s_2, \ldots, s_{K-1}$) and voicing detection (read from state s_0) are disentangled from each other. In the proposed regression approach, we require different strategies. In this section, we focus on F0 tracking errors only; a comparison of voicing detectors in the classification and regression approaches will be provided later in Section V.

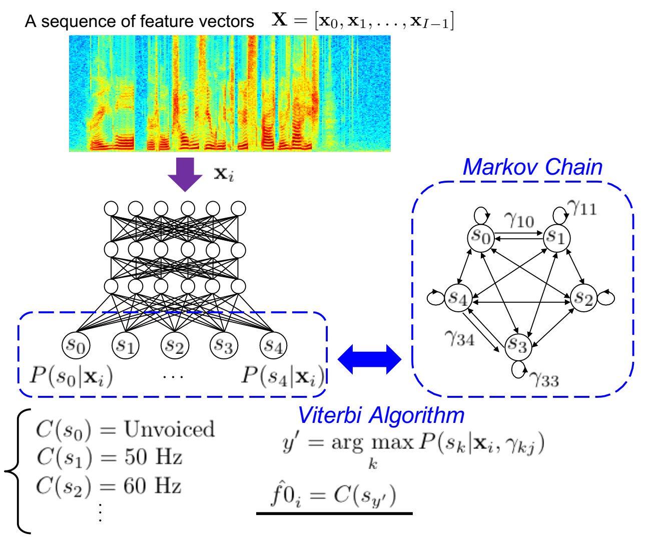
A sequence of feature vectors $\mathbf{X} = [\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{I-1}]$
$\mathbf{x}_i$
Markov Chain
γ_{10}
γ_{11}
γ_{34}
γ_{33}
s_0 s_1 s_2 s_3 s_4
$P(s_0|\mathbf{x}_i)$ $\cdots$ $P(s_4|\mathbf{x}_i)$
$C(s_0) = \text{Unvoiced}$
$C(s_1) = 50$ Hz
$C(s_2) = 60$ Hz
Viterbi Algorithm
$y' = \arg\max_k P(s_k|\mathbf{x}_i, \gamma_{kj})$
$\hat{f0}_i = C(s_{y'})$

Fig. 1. An overview of a DNN-based classification approach to F0 estimation. A DNN is trained to derive the posterior probabilities of each state given input features. The quantized frequency associated with the most likely state, $C(s_{y'})$, represents the estimate of F0 corresponding to the input features, $\mathbf{x}_i$. Voicing detection is achieved by representing the unvoiced frames using an additional state, s_0.

B. Estimation Error

The estimation error from the classification task depends on how the frequency space is quantized. In this section, we analyze the relation between the quantization settings (*i.e.*, the number of states) and the F0 estimation error.

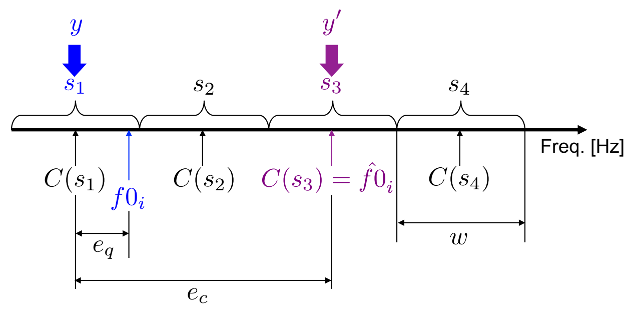

Fig. 2. F0 estimation error of the classification approach is determined by the sum of the quantization error and the classification error. In this case, e_q has a negative value.

The F0 estimation error at the i-th frame, ϵ_i, can be measured by [31]

$$\epsilon_i = \left| \hat{f0}_i - f0_i \right|, \tag{3}$$

where $f0_i$ and $\hat{f0}_i$ denote, respectively, the ground truth and the estimated F0 values in Hz at frame i. We assume that ϵ_i comprises the *quantization error* and the *classification error*. The former is the difference between the ground truth value, $f0_i$, and the quantized value, $C(s_y)$, corresponding to the frequency state to which the ground truth belongs. Let s_y represent the frequency interval from f_1^y to f_2^y and let $C(s_y)$ represent the median of the segment:

$$C(s_y) = \frac{f_1^y + f_2^y}{2}. \tag{4}$$

Therefore, the quantization error, e_q, is derived as follows:

$$e_q = C(s_y) - f0_i = \frac{f_1^y + f_2^y}{2} - f0_i. \tag{5}$$

The classification error, e_c, is the difference between the quantized values representing the most likely state, $C(s_{y'})$, and $C(s_y)$. In this work, we divide the frequency space uniformly into frequency states, *i.e.*, the same interval, w, with no overlap. Therefore, e_c is defined as

$$e_c = C(s_{y'}) - C(s_y) = w\,(y' - y)\,, \tag{6}$$

where

$$w = f_2^y - f_1^y = f_2^k - f_1^k, \qquad \forall k \in \{0, 1, \ldots, K-1\}. \tag{7}$$

Consequently, the estimation error is the sum of e_q and e_c,

$$\epsilon_i = |e_q + e_c| \tag{8}$$
$$= \left| \frac{f_1^y + f_2^y}{2} - f0_i + (f_2^y - f_1^y)\,(y' - y) \right| \tag{9}$$

Fig. 2 illustrates the quantization and classification errors.

Now, we assume that the probability density of $f0_i$ within each interval of frequency states, $P(f0_i)$, is uniform, as shown in Fig. 3, and e_q at each $f0_i$ is represented in Fig. 4. In this

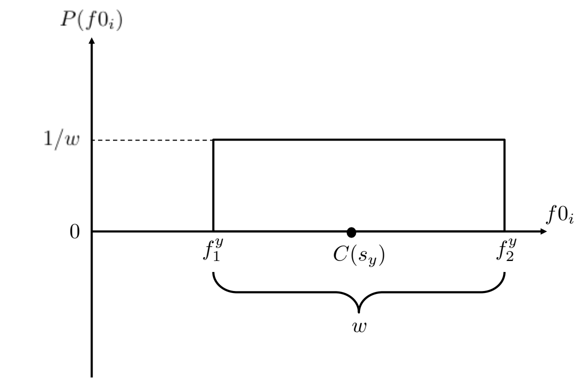

Fig. 3. We assume that the probability density, $P(f0_i)$, within a quantized F0 state, s_y, is uniform. Therefore, $P(f0_i)$ is equal to $1/w$ at any $f0_i$ within a state.

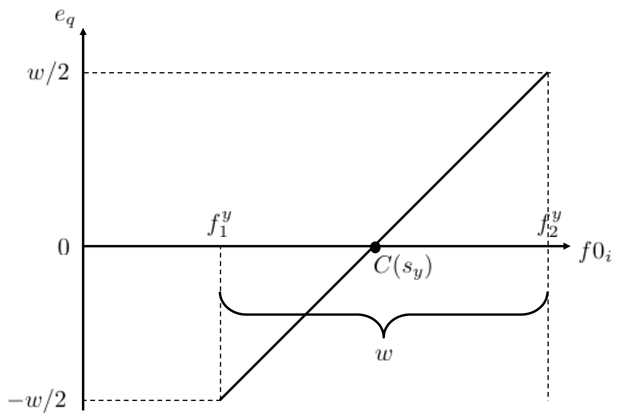

Fig. 4. The quantization error, e_q, is determined by the value of $f0_i$ within the range between $-w/2$ and $w/2$.

case, the expected value of e_q is defined as

$$\mathbb{E}\left[e_q\right] = \int_{f_1^y}^{f_2^y} P(f0_i) e_q \, \mathrm{d}f0_i = \frac{1}{w}\left(2\int_0^{w/2} f0_i \, \mathrm{d}f0_i\right) = \frac{w}{4} \tag{10}$$

where $\mathbb{E}[\cdot]$ denotes the expectation. Consequently, e_q depends on w: a smaller w (the more frequency states with higher resolution) leads to lower quantization errors. On the other hand, e_c could depend on both w and $f0_i$. Therefore, the probability density of e_c can vary dynamically, and $\mathbb{E}[e_c]$ cannot be analytically derived.

C. Empirical Analysis of Estimation Error

Since we cannot calculate $\mathbb{E}[e_c]$, we analyze e_c, e_q and ϵ_i empirically. To this end, we first train a four-layer (*i.e.*, three hidden layers), *fully connected feedforward deep neural network* (FC-DNN), which is widely used for classification approaches to F0 estimation [20]. The training dataset includes multispeaker speech and the ground truth of the F0 values. The details of the dataset and the FC-DNN setup are provided in Section IV-A.

After the classification tests with the FC-DNNs, ϵ_i, e_q and e_c are calculated with Equations (3), (5) and (6), respectively. Fig. 5 demonstrates how these errors change with the number of states, *i.e.*, the length of w.

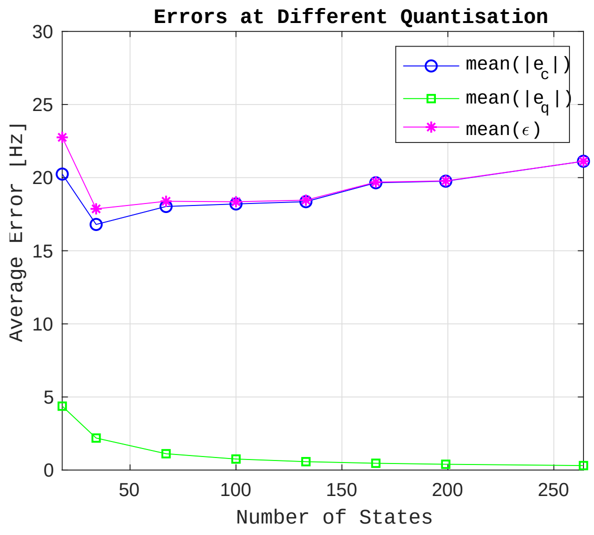

Fig. 5. The averages of the classification errors, mean($|e_c|$), quantization errors, mean($|e_q|$), and estimation errors, mean(ϵ), at different numbers of the frequency states.

As expected, e_q is reduced at more output states, as represented in Equation (10). Therefore, we can reduce the quantization error by applying finer quantization. On the other hand, e_c shows an increase after 34 states. Consequently, at 166 states and beyond, the deterioration in e_c exceeds the advantage in e_q, and the total estimation error increases. Fig. 6 illustrates F0 contours of same speech corresponding to the word “DARK” with different quantization settings, where each contour is shifted by 5 Hz for better visualization. Even at 133 states, the F0 contour remains discontinuous due to the quantization, and we expect deterioration in the estimation error at more frequency states.

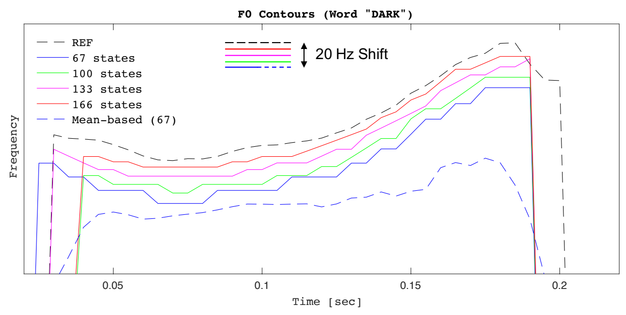

Fig. 6. F0 contours of same speech corresponding to the word “DARK” with each quantization settings. Each contour is shifted by 5 Hz for better visualization. The blue dashed line represents the smoothed contour of a 67-state classification with mean-based generation.

Some methods exist for generating a smoothed contour from quantized F0 values to alleviate the impact of quantization. In particular, *mean-based generation* is common in speech synthesis (*e.g.*, [23]). Instead of selecting the F0 state with the maximum posterior probability, mean-based generation outputs a smoothed F0 value according to

$$\widehat{f0}_i = \sum_{k=1}^{K-1} C(s_k) P(s_k|\mathbf{x}_i), \tag{11}$$

which works well if the softmax distribution at the output of the DNN is sharply peaked (*i.e.*, $P(s_k|\mathbf{x}_i) \approx 0$ outside of some narrow interval $k \in [k^* - \delta, k^* + \delta]$ centered around state k^* with small δ). This condition might hold in speech synthesis where linguistic features are mapped to F0 values. In F0 tracking for noisy speech, however, we typically observe multiple distinct local maxima due to harmonic structures and ambiguity in F0 in speech, as shown in Fig. 7, illustrating a softmax distribution for 30 consecutive frames of speech. This

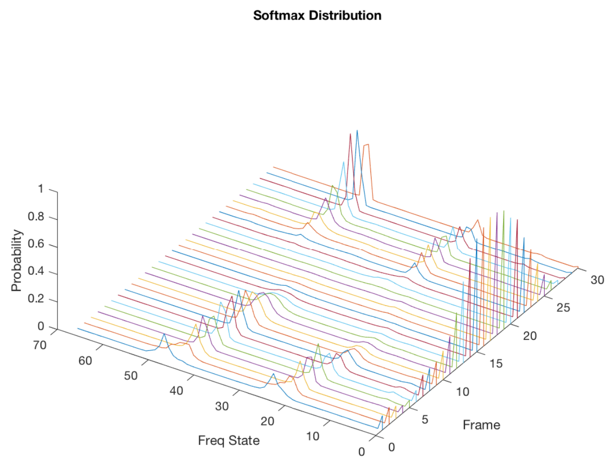

Fig. 7. The softmax distributions for 30 frames of speech, illustrating multiple local peaks corresponding to harmonic frequencies.

phenomenon results in a smeared F0 track and blunt edges at the voiced/unvoiced boundaries, as shown in Fig. 6.

In summary, one cannot eliminate the problem of frequency quantization from the classification approach by simply increasing the number of F0 states or by mean-based smoothing. This issue motivates us to study F0 estimation using RNN-based regression models.

III. RNN-Based Regression Approaches to F0 Estimation

In this section, we describe our two proposed F0 estimators. The first estimator uses an RNN regression model to map time-frequency features onto F0 values, while the second estimator uses an RNN-based waveform-to-sinusoid regression model.

A. Proposed Method 1: Spectrum-to-F0 Regression

In our preliminary study [15], we found that RNN-based regression models achieve better performance in mapping speech features in the time-frequency domain onto F0 than feedforward DNNs. Therefore, we restrict ourselves to RNN-based regression models only. Our method consists of two steps, F0 determination and voicing decision, which are detailed in the following two sub-subsections.

1) F0 determination: F0 determination is achieved by RNN modeling of raw magnitude spectra, in which F0 and its harmonic multiples are often clearly visible. The RNN part, in turn, benefits from the temporal dependency of F0 across consecutive frames; it can assist in F0 estimation even if a specific frame might display less clear harmonics. To this end, the discrete time-domain speech signal, $u(n)$, is first divided into I frames, $u_0(m), u_1(m), \ldots, u_{I-1}(m)$, where m denotes the local time index within a frame. Each frame is transformed to the frequency domain by the *short-time Fourier transform* (STFT) to obtain a time sequence of magnitude spectra, $\mathbf{X}$:

$$\begin{aligned} \mathbf{X} &= [\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{I-1}] \\ \mathbf{x}_i &= [|x_i(0)|, |x_i(1)|, \ldots, |x_i(P-1)|]^\top \\ x_i(p) &= \mathcal{F}\{u_i(m)\} \equiv \sum_{m=0}^{M-1} \varphi(m) u_i(m) e^{-\mathrm{j}2\pi mp/M}, \end{aligned} \tag{12}$$

where $\mathcal{F}\{\cdot\}$ denotes windowed *discrete Fourier transform* (DFT) operator with j, $\varphi(\cdot)$ and M being the imaginary unit, window function and zero-padded frame size, respectively. Furthermore, $P = M/2 + 1$ denotes the number of DFT bins between 0 Hz and the Nyquist frequency of $u(n)$. A subsequence of consecutive frames, centered around frame i, is extracted from $\mathbf{X}$ as an input to the RNN. In the following description, we use $\mathbf{X}[i:j]$ to indicate a subsequence starting from frame i and ending with frame j.

Each layer in the RNN has feedback connections for sending the output back to its own input in addition to the feed-forward connections to the next layer. Therefore, an RNN layer receives its own output at the previous time sequence as well as the current time sequence input from the previous layer. This behavior of RNN layers, which are interpreted as memory cells, demonstrates the temporal dependency of the input sequence by capturing temporal dynamics of speech. Such basic RNN cells are, however, incapable of learning long-term dependencies such as the whole length of an utterance [32]. Therefore, *long short-term memory* (LSTM) networks [33], in which the cell states are selectively controlled by the input, output and forget gates to sustain long-term dependency [34]–[36], have become commonly used in many applications. When the future information as well as the current and past information of the input sequence is known, many recent applications commonly use *bidirectional* long short-term memory (BLSTM) [37] to capture global dependencies [38]–[40]. In our preliminary experiments, we examined BLSTM applied to a full sequence $\mathbf{X}[0:I-1]$, leading to insufficient accuracy[1]. We assume that the local dependency between neighboring frames is more important for F0 tracking. Therefore, we model subsequences of length $2q+1$, namely, $\mathbf{X}[i-q:i+q]$, rather than the full sequence, to encode *local dependency*.

Our RNN regression model is represented by a *sequence-to-scalar encoder* structure, as illustrated in Fig. 8. The output

[1]In these preliminary experiments, the gross pitch error rate for clean speech was 54.2 % and 50.5 % for the whole-sequence LSTM and BLSTM, respectively. One may require more complex structures such as a BLSTM encoder-decoder framework including an attention mechanism for the F0 tracking task to benefit from the global long-term dependency. However, this exploration is beyond the scope of this paper.

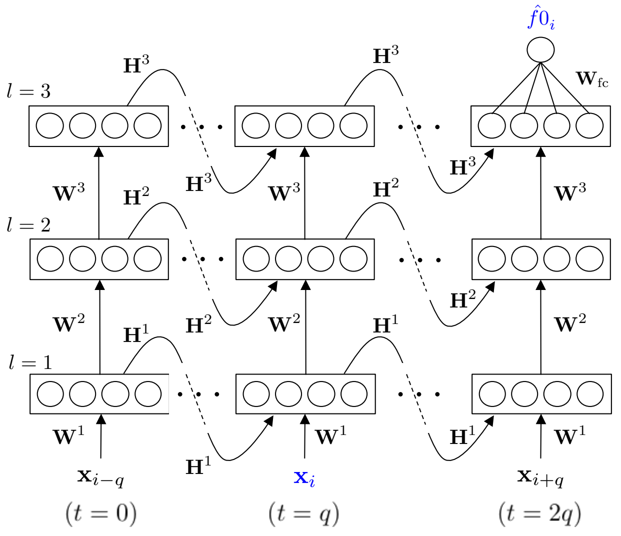

Fig. 8. An unrolled diagram showing the RNN regression model for F0. The model is formed with an encoder structure where an sequence of input vectors are encoded to a vector (or scalar) at the final time step.

at an RNN layer l at a time sequence, t $(t = 0, 1, \ldots, 2q)$, where $\boldsymbol{\theta}_t^l$, is derived as follows with respect to the analyzed frame, $\mathbf{x}_i$.

$$\begin{aligned} \boldsymbol{\theta}_t^l &= g\left(\mathbf{W}^l \boldsymbol{\phi}_t^l + \mathbf{H}^l \boldsymbol{\theta}_{t-1}^l\right) & (13) \\ \boldsymbol{\phi}_t^l &= \left[1, (\boldsymbol{\theta}_t^{l-1})^\top\right]^\top & (14) \\ \boldsymbol{\theta}_t^0 &= \left[1, (\mathbf{x}_{i-q+t})^\top\right]^\top & (15) \\ \boldsymbol{\theta}_{-1}^l &= \mathbf{0}. & (16) \end{aligned}$$

Here, $g(\cdot)$ represents an activation function and $\mathbf{W}^l$ and $\mathbf{H}^l$ are weight matrices at layer l consisting of feedforward and feedback weights as follows.

$$\mathbf{W}^l = \begin{bmatrix} w_{10}^l & w_{11}^l & \cdots & w_{1a_{(l-1)}}^l \\ w_{20}^l & w_{21}^l & \cdots & w_{2a_{(l-1)}}^l \\ \vdots & \vdots & \ddots & \vdots \\ w_{a_l 0}^l & w_{a_l 1}^l & \cdots & w_{a_l a_{(l-1)}}^l \end{bmatrix} \quad (17)$$

$$\mathbf{H}^l = \begin{bmatrix} h_{11}^l & h_{12}^l & \cdots & h_{1a_l}^l \\ h_{21}^l & h_{22}^l & \cdots & h_{2a_l}^l \\ \vdots & \vdots & \ddots & \vdots \\ h_{a_l 1}^l & h_{a_l 2}^l & \cdots & h_{a_l a_l}^l \end{bmatrix} \quad (18)$$

where w_{jk}^l denotes the feedforward weight from the k-th unit in layer $l-1$ to the j-th unit in layer l (the 0-th unit is the bias unit), while h_{jk}^l is the feedback weight from the k-th unit to the j-th unit within layer l. a_l denotes the number of units in layer l.

Only the last time sequence $(t = 2q)$, has an output layer consisting of a single unit connected with the previous RNN layer with a feedforward weight matrix, $\mathbf{W}_{\text{fc}}$, to output the estimate of $f0_i$ as a sequence-to-scalar transform:

$$y = \mathcal{I}\left(\mathbf{W}_{\text{fc}} \boldsymbol{\phi}_{2q}^L\right) = \hat{f0}_i, \quad (19)$$

where $\mathcal{I}(\cdot)$ is the identity function to activate the output layer and L denotes the number of RNN layers.

Weight optimization is accomplished with minibatch gradient descent with the backpropagation algorithm [41] in which the regression target is set to the ground-truth F0 value for voiced frames and zero for unvoiced frames.

2) Voicing determination: An important merit of the classification approach is the simplicity of voicing detection noted in Section II: voicing prediction is read directly from the additional state of the network output. We would similarly like to utilize directly the output of our spectrum-to-F0 regression model, *i.e.*, the estimated F0 value, $\hat{f0}_i$, for voicing detection. One challenge is that, unlike in the classification approach where F0 tracking and voicing detection are disentangled from each other, $\hat{f0}_i$ resides along the *same* continuous real line $\mathbb{R}$ (including possibly negative values), regardless of whether the frame is unvoiced or voiced. As unvoiced frames lack periodic structure, their F0 is, by definition, *undefined*. For this reason, F0 for unvoiced frames is generally modeled in isolation from the continuous distribution of F0 for voiced frames in methods employed in speech synthesis tasks [26].

In practice, we can use estimated F0 values for voicing detection by assuming that the values for unvoiced frames are concentrated at lower values than those for voiced frames. We have seen that this assumption holds well, which suggests a simple voicing detector based solely on the estimated F0 value: one declares a frame as voiced when the estimated F0 exceeds a certain threshold. The immediate question, however, is how to make such a rule speaker-independent: recall that the F0 range greatly varies between genders and between individuals, which suggests adapting the threshold based on the overall pitch range of the speaker.

Our proposed *bi-Gaussian voicing detector*, detailed below, formalizes this idea. It is inspired by implicit voicing condition modeling with a continuous F0 distribution in text-to-speech applications [26] and from similar ideas used in *speech activity detection* (SAD) [27], [28] — another frame-level binary classification problem — to model short-term energy distributions of non-speech and speech frames. In addition to being speaker-adaptive, our approach has another favorable property of being completely *unsupervised* — it requires neither offline training data nor voiced/unvoiced labels. For these reasons, the method can be applied to voicing detection of any F0 tracker that fulfills the above-noted ordering property.

Voicing detection can be cast as a decision-theoretic problem. Depending on the application, the interest is in either a *soft* decision (*e.g.*, voicing probability) or a *hard* binary decision, obtained by thresholding a soft indicator function. Given the observed F0 value at the i-th frame, $\hat{f0}_i$, the task is to choose a optimal class $v_i \in \mathcal{H}$ from a hypothesis space $\mathcal{H} = \{\mathcal{H}_0, \mathcal{H}_1\}$. Here, $\mathcal{H}_0$ and $\mathcal{H}_1$ denote, respectively, voiced and unvoiced states[2], and optimality refers to a decision policy

[2]In many F0 trackers, it is a standard convention to represent unvoiced frames using a special F0 value of 0. For the reader who may wonder why we chose to use '0' to indicate a voiced state, we follow the standard convention in statistical hypothesis testing to denote the positive class (here, voiced speech) as $\mathcal{H}_0$. The two exhaustive and mutually exclusive hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$ are known, respectively, as the *null* and the *alternative* hypotheses. By convention, higher LLR scores indicate stronger support in favor of accepting $\mathcal{H}_0$ (and rejecting $\mathcal{H}_1$).

to minimize the empirical Bayes risk [42]. Specifically, we choose the class with the higher posterior probability:

$$\begin{aligned} v_i &= \arg\max_{j=0,1} P(\mathcal{H}_j|\hat{f}0_i) \\ P(\mathcal{H}_j|\hat{f}0_i) &= \frac{p(\hat{f}0_i|\mathcal{H}_j)\, P(\mathcal{H}_j)}{p(\hat{f}0_i)}, \end{aligned} \tag{20}$$

which minimizes the probability of voicing error, assuming that all the probabilistic quantities are known. Here, $P(\mathcal{H}_j)$ and $p(\hat{f}0_i|\mathcal{H}_j)$ are, respectively, the *prior probability* and the *likelihood function* of class j, while $p(\hat{f}0_i)$ is a normalizer to ensure $P(\mathcal{H}_0|\hat{f}0_i) + P(\mathcal{H}_1|\hat{f}0_i) = 1$; similarly, $P(\mathcal{H}_0) + P(\mathcal{H}_1) = 1$. The decision rule (20) can be rewritten in an alternative form:

$$\ell(\hat{f}0_i) \underset{\texttt{unvoiced}}{\overset{\texttt{voiced}}{\gtrless}} \eta, \tag{21}$$

where $\ell(\hat{f}0_i) := \log p(\hat{f}0_i|\mathcal{H}_0) - \log p(\hat{f}0_i|\mathcal{H}_1)$ is a *log-likelihood ratio* (LLR) score, containing all the data-related terms, and $\eta := -\text{logit}\, P(\mathcal{H}_1)$ is a fixed decision threshold[3]. We fix $\eta = 0$ (*i.e.*, assume that $P(\mathcal{H}_0) = P(\mathcal{H}_1)$) to reflect the maximum a priori uncertainty about the class. In our model, we train a parametric likelihood ratio detector $\ell(\hat{f}0_i|\boldsymbol{\theta}_r)$ for each recording (or a set of recordings), r, using only the F0 values of that recording. Therefore, a fixed threshold is not restrictive, as the left-hand side of (21) is adapted to the F0 distribution of r.

We model each state in the recording r as a Gaussian likelihood with a shared variance parameter, σ_r^2:

$$\begin{aligned} p(\hat{f}0_i|\mathcal{H}_0, r) &= \mathcal{N}(\hat{f}0_i|\, \mu_{r,0}, \sigma_r^2) \\ p(\hat{f}0_i|\mathcal{H}_1, r) &= \mathcal{N}(\hat{f}0_i|\, \mu_{r,1}, \sigma_r^2), \end{aligned} \tag{22}$$

where $\mathcal{N}(\cdot)$ denotes the Gaussian likelihood and $\mu_{r,j}$ is the mean F0 of class $\mathcal{H}_j$ in r. The assumption of a shared variance parameter may look restrictive but was empirically found to outperform class-specific variances in our preliminary experiments. An intuitively appealing implication is that the decision rule in (21) leads to a thresholding rule in which a frame is declared voiced if the estimated F0 exceeds a threshold, which is generally not the case for unconstrained variances.[4] Specifically, by substituting (22) into (21) with $\eta = 0$, we obtain an equivalent voicing detector for recording r:

$$\hat{f}0_i \underset{\texttt{unvoiced}}{\overset{\texttt{voiced}}{\gtrless}} \frac{1}{2}\big(\mu_{r,0} + \mu_{r,1}\big), \tag{23}$$

where the expression on the right-hand side coincides with the crossover point of the two Gaussians. Since both $\mu_{r,0}$ and $\mu_{r,0}$ depend on the speaker (and his or her gender), the threshold on the right-hand side of (23) is adapted according to the overall F0 range of the speaker.

The above formulation requires knowledge of the class-conditional parameters (or their estimates). As the true voicing labels of r are unavailable, we estimate the parameters $\boldsymbol{\theta}_r = (\mu_{r,0}, \mu_{r,1}, \sigma_r^2)$ via a standard *expectation-maximization* (EM) algorithm [45] by fitting a 2-component *Gaussian mixture model* (GMM) to the F0 observations of r. The two estimated Gaussians are assumed to be ordered so that the one with the lower mean represents unvoiced frames and the other represents voiced frames.

[3] $\text{logit} P := \log P - \log(1 - P)$

[4] This intimately relates to the *calibration* of log-likelihood ratios; the topic is beyond the scope of this work, but we point the interested reader to [43], [44] for a discussion in the context of automatic speaker verification.

B. Proposed Method 2: Waveform-to-Sinusoid Regression

Our second method for F0 tracking is based on the premise that extraction of the F0 value from a pure (noisy) sinusoid is straightforward through an autocorrelation approach. Since observed human speech, however, is *not* sinusoidal but contains multiple frequency components (including aperiodic ones) resulting from glottal source waveform shapes, vocal tract filtering, background noise and other factors, we need a preprocessing approach to suppress these nuisance variables. To this end, our second approach consists of three steps. First, an input speech waveform is transformed to a single sinusoid encoding F0 with RNN-based regression. Second, F0 is estimated from this approximated sinusoid using a simple autocorrelation approach. Finally, voicing decisions are made in a similar manner as in our first approach. These three steps are detailed in the following sub-subsections.

1) Waveform-to-sinusoid regression: In the first step, to transform a waveform into a sinusoid, framed time-domain speech waveforms, $u_0(m), u_1(m), \ldots, u_{I-1}(m)$, are directly used as input to an RNN. As the input is a voiced frame, the outputs of the RNN are mapped onto a single sinusoid oscillated with F0 of the input waveform. Otherwise, for unvoiced and nonspeech inputs, the RNN maps the input onto the input itself as an identity mapping. The estimate of F0 is then explicitly inferred from the resultant single sinusoid using its autocorrelation. Fig. 9 illustrates the proposed framework.

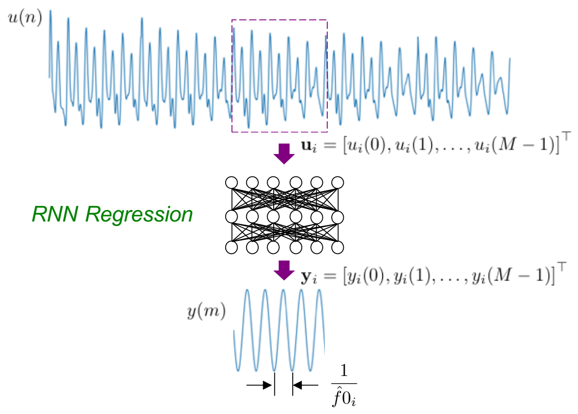
$u(n)$
$\mathbf{u}_i = [u_i(0), u_i(1), \ldots, u_i(M-1)]^\top$
RNN Regression
$\mathbf{y}_i = [y_i(0), y_i(1), \ldots, y_i(M-1)]^\top$
$y(m)$
$\frac{1}{\hat{f}0_i}$

Fig. 9. A voiced speech waveform directly inputs to an RNN to perform waveform-to-sinusoid regression. The estimate of F0 is then inferred from the resultant sinusoid.

The speech signal $u(n)$ is first divided into I frames, $\mathbf{u}_0$, $\mathbf{u}_1$, ..., $\mathbf{u}_{I-1}$:

$$\mathbf{u}_i = [u_i(0), u_i(1), \ldots, u_i(M-1)]^\top, \tag{24}$$

where M denotes the number of samples in a frame. The waveform-to-sinusoid regression model also takes a sequence-to-vector encoding structure similar to the RNN regression approach described in Section III-A. Therefore, the output at the RNN layer l at the time sequence t $(t = 0, 1, \ldots, 2q)$, $\boldsymbol{\theta}_t^l$, is derived as follows with respect to the analyzed frame, $\mathbf{u}_i$.

$$\boldsymbol{\theta}_t^l = g\left(\mathbf{W}^l \boldsymbol{\phi}_t^l + \mathbf{H}^l \boldsymbol{\theta}_{t-1}^l\right) \quad (25)$$

$$\boldsymbol{\phi}_t^l = \left[1, (\boldsymbol{\theta}_t^{l-1})^\top\right]^\top \quad (26)$$

$$\boldsymbol{\theta}_t^0 = \left[1, (\mathbf{u}_{i-q+t})^\top\right]^\top \quad (27)$$

$$\boldsymbol{\theta}_{-1}^l = \mathbf{0}, \quad (28)$$

where $g(\cdot)$ represents an activation function and $\mathbf{W}^l$ and $\mathbf{H}^l$ are weight matrices with the same forms as Equations (17) and (18), respectively.

To achieve the waveform-to-sinusoid regression, the output layer is activated by the identity function. Consequently, the outputs of the RNN at the i-th frame (i.e., $t = q$), $\mathbf{y}_i$, are derived as follows:

$$\mathbf{y}_i = \mathcal{I}\left(\mathbf{W}^L \boldsymbol{\phi}_{2q}^L + \mathbf{H}^L \boldsymbol{\theta}_{2q-1}^L\right) \quad (29)$$

$$= \left[y_i(0), y_i(1), \ldots, y_i(M-1)\right]^\top, \quad (30)$$

where L denotes the number of RNN layers and $y_i(m)$ is a discrete time-domain sinusoid which is oscillated with $\hat{f0}_i$.

In the training process, $\mathbf{W}^l$ and $\mathbf{H}^l$ are optimized in advance by supervised learning. During voiced speech periods, the weight parameters are trained by minimizing the mean square error (MSE) between $\mathbf{y}_i$ and the target sinusoid. As illustrated in Fig. 10, the target sinusoid is oscillated with $f0_i$ and φ_i, which are the ground truth of F0 at frame i and the phase to maximize the cross-correlation between $\mathbf{u}_i$ and $\cos(2\pi f0_i m/f_s)$, respectively. Here, f_s is the sampling frequency and $m = 0, 1, \ldots, M-1$.

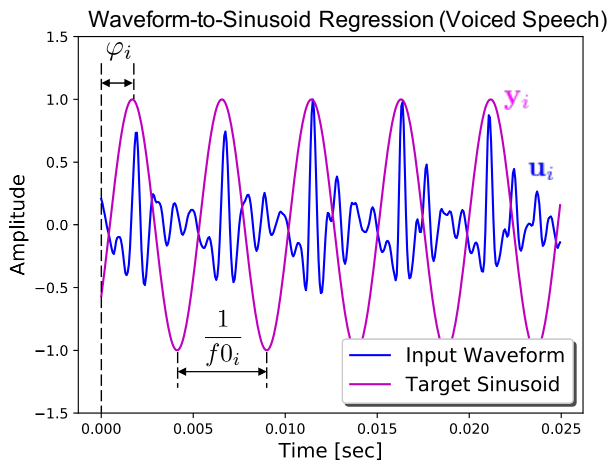

Fig. 10. The target sinusoid for supervised learning, $\mathbf{y}_i$, is oscillated with the ground truth of F0 at frame i $(f0_i)$.

Weight optimization during training is accomplished with minibatch gradient descent with the backpropagation algorithm [41]. Fig. 11-(a) illustrates the relation between an input voiced waveform, $\mathbf{u}_i$, and an output sinusoid, $\mathbf{y}_i$, at different SNRs for white noise contaminated speech. Fig. 11-(b) depicts their magnitude spectra to demonstrate how $\mathbf{y}_i$ is more suitable for F0 analysis than $\mathbf{u}_i$.

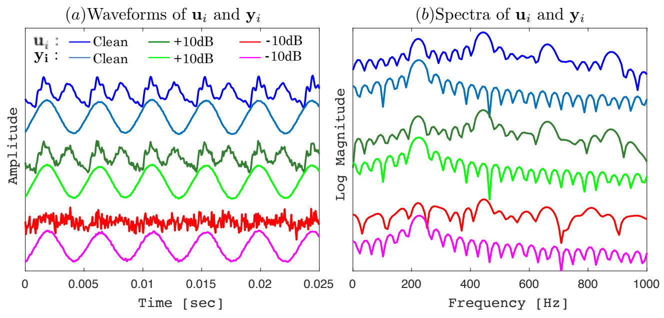

Fig. 11. (a) depicts the relation between $\mathbf{u}_i$ and $\mathbf{y}_i$ in clean and noisy conditions (SNRs between +10 dB and -10 dB for white noise). (b) plots their magnitude spectra. The waveforms and spectra in each plot have been shifted for better visualization.

2) F0 determination from the sinusoid: In the second step, to estimate F0 at the i-th frame, $\hat{f0}_i$ is inferred from $\mathbf{y}_i$ by maximizing its autocorrelation.

$$\hat{f0}_i = \frac{f_s}{\hat{\tau}_i} \quad (31)$$

$$\hat{\tau}_i = \arg\max_m \mathcal{R}_{\mathbf{yy}}, \quad (32)$$

where $\mathcal{R}_{\mathbf{yy}}$ denotes the autocorrelation of $\mathbf{y}_i$, f_s is the sampling frequency of the input waveform and m is sample shift to calculate $\mathcal{R}_{\mathbf{yy}}$.

3) Voicing determination: The final step is the voicing decision. In a preliminary experiment, we tried the same bi-Gaussian approach to model the distribution of the estimated F0 values. This approach, however, leads to unacceptably high voicing error rates ($\sim 40\%$), which is close to the level expected by chance (50 %) in clean data. In our first method in Section III-A, the F0 value (with a value of 0 assigned to current unvoiced frames) is directly used as the RNN regression target, whereas our second method encodes F0 indirectly to the sinusoid, represented by a vector. We observed that the estimated F0 values from voiced and unvoiced frames highly overlap. Therefore, a revised voicing detector that uses other characteristics of the sinusoid was deemed necessary. The voicing decision rule is formed from a bivariate extension of the same approach as that described above.

We first extract the voicing feature vector, $\mathbf{a}_i$, at the i-th frame from $\mathbf{y}_i$. From the waveform-to-sinusoid regression process, we can find two different characteristics in $\mathbf{y}_i$ between the voiced and unvoiced frames. First, voiced waveforms are mapped onto sinusoids, while unvoiced waveforms are mapped onto themselves, which are not sinusoids. Second, the amplitude of the target sinusoid for the voiced frames is 1, while the amplitude of the target waveform for the unvoiced frames are generally much less than 1, *i.e.*, identical to the amplitude of the input unvoiced waveform. From these characteristics of $\mathbf{y}_i$, we assume that a voicing feature vector, $\mathbf{a}_i$ can be an effective input of the bi-Gaussian voicing detector as a two-dimensional vector representing the variance of $\mathbf{y}_i$

and the similarity of $\mathbf{y}_i$ to a sinusoid. Therefore, we determine the input of the bi-Gaussian voicing detector $\mathbf{a}_i$:

$$\begin{aligned} \mathbf{a}_i &= \left[\sigma_{yi}^2, r_{yi}\right] && (33)\\ r_{yi} &= \frac{\operatorname{cov}\left(\mathbf{y}_i, \bar{\mathbf{y}}_i\right)}{\sigma_{yi}\sigma_{\bar{y}i}} && (34)\\ \bar{\mathbf{y}}_i &= \cos\left(2\pi \hat{f0}_i \frac{m}{f_s} + \varphi_i\right) && (35)\\ \varphi_i &= 2\pi \hat{f0}_i \frac{m_0}{f_s} && (36)\\ m_0 &= \arg\max_m \mathcal{R}_{y\bar{y}} && (37) \end{aligned}$$

where $\operatorname{cov}(\mathbf{y}_i, \bar{\mathbf{y}}_i)$ and $\mathcal{R}_{y\bar{y}}$ denote the covariance and cross-correlation of $\mathbf{y}_i$ and $\bar{\mathbf{y}}_i$ and σ_{yi}^2 represents the variance of the values in $\mathbf{y}_i$. Cosine wave, $\bar{\mathbf{y}}_i$, plays a role as a reference sinusoid to gauge the similarity of $\mathbf{y}_i$ to a sinusoid when it has the same frequency and phase as $\mathbf{y}_i$. Therefore, the frequency and phase of $\bar{\mathbf{y}}_i$ are determined by $\hat{f0}_i$ and m_0 respectively.

The 2-dimensional voicing feature vector, $\mathbf{a}_i \in \mathbb{R}^2$, is then modeled using bivariate Gaussians,

$$\begin{aligned} p(\mathbf{a}_i|\mathcal{H}_0, r) &= \mathcal{N}(\mathbf{a}_i|\,\boldsymbol{\mu}_{r,0}, \boldsymbol{\Sigma}_{r,0})\\ p(\mathbf{a}_i|\mathcal{H}_1, r) &= \mathcal{N}(\mathbf{a}_i|\,\boldsymbol{\mu}_{r,1}, \boldsymbol{\Sigma}_{r,1}), \end{aligned} \qquad (38)$$

where $\boldsymbol{\mu}_{r,0} \in \mathbb{R}^2$ and $\boldsymbol{\mu}_{r,1} \in \mathbb{R}^2$ are the mean vectors of the voiced and unvoiced frames, respectively, and $\boldsymbol{\Sigma}_{r,0} \in \mathbb{R}^{2\times2}$ and $\boldsymbol{\Sigma}_{r,1} \in \mathbb{R}^{2\times2}$ are full covariance matrices of the voiced and unvoiced classes. Different from the voicing detector in our first method, here we use class-dependent, rather than shared covariances. This is an experimental choice that better exploits class differences in terms of covariances, as demonstrated in Section V-C. The parameters $\boldsymbol{\theta}_r = (\boldsymbol{\mu}_{r,0}, \boldsymbol{\mu}_{r,1}, \boldsymbol{\Sigma}_{r,0}, \boldsymbol{\Sigma}_{r,1})$ are again trained in an unsupervised way via the EM algorithm and with the LLR criterion applied at a threshold of 0 to perform voicing detection. However, unlike in the univariate case, technically speaking, we cannot sort vectors from lower to higher values. Recalling that the voiced class is represented by higher values in both dimensions of $\mathbf{a}_i$, we designate the Gaussian with the higher Euclidean norm of its mean, $\|\boldsymbol{\mu}_{r,\cdot}\|_2$, as the voiced class.

IV. Experimental Settings

In our experiments, we compare the performance of the following eight F0 trackers in terms of their accuracy and noise robustness:

Proposed regression methods:

- REG(sp2f0): Spectrum-to-F0 regression described in Section III-A
- REG(wav2sin): Waveform-to-Sinusoid regression described in Section III-B

State-of-the-art classification methods:

- CLS(sp2f0) [20]
- CLS(wav2f0) [22]
- CREPE [13]

Signal processing methods:

- RAPT [9]
- pYIN [11]
- PEFAC [14]

CLS(sp2f0) is a DNN classification model from *spectrum* to quantized frequency states, whereas CLS(wav2f0) employs a DNN classification model that maps a *waveform* onto quantized frequency states. These classification models use fully connected feedforward networks with three hidden layers and 1024 units, since it has been reported that the recurrent architecture of their models do not yield substantial improvement [20], [22]. Table II summarizes their network architectures. CREPE is an F0 tracker for music signals. It uses a CNN-based classification model that consists of six convolutional layers and a fully connected layer to map waveforms onto quantized frequency states.

TABLE II
Settings for the FC-DNNs of CLS(sp2f0) and CLS(wav2f0)

Parameter	Input Layer	Hidden Layers	Output Layer
# of Layers	1	3	1
# of Units	CLS(sp2f0): 3591 CLS(wav2f0): 400	1024	68
Activation	-	ReLU	Softmax
Minibatch size	200		

A. Datasets (PTDB-TUG Corpus)

We use the *pitch tracking database from Graz University of Technology* (PTDB-TUG) [46] to compare the above F0 trackers. As summarized in Table III, for the machine learning techniques, we use a **training set** of 3200 utterances spoken by 16 speakers (8 men and 8 women), *i.e.*, 200 utterances for each speaker, and a **validation** dataset of 574 utterances spoken by the same 16 speakers (36 utterances for each speaker except speaker "F08", who provided 34 utterances). All the F0 trackers were compared using a **test set** of 944 utterances spoken by 4 speakers (2 men and 2 women), which was disjoint from the training and validation set speakers.

TABLE III
Data allocation from PTDB-TUG to the datasets

Subset	Speakers	Utts (per speaker)	Duration
Training	8 Females + 8 Males	3,200 (200)	372 min
Validation	8 Females + 8 Males	574 (36)*	62 min
Test	2 Females + 2 Males (Unknown Speakers)	944 (236)	104 min

*34 utterances for speaker "F08"

The PTDB-TUG corpus contains ground-truth F0 values obtained from *laryngograph* signals recorded in clean conditions, to which a Kaiser filter and RAPT are applied. We used these values in the following experiments as the ground truth.

B. Noisy Conditions (NOISEX-92)

The speech utterances in each dataset are sampled at 16 kHz, and the sampled signals in the training and validation sets were contaminated with eight types of additive noise at five levels of the global SNRs (-10, -5, 0, +5, and +10 dB). Eight types of **known noises**, displayed in Table IV, are included in all three data subsets. Another set of four **unknown noises** are included in the test set only.

In summary, the training set contains 131,200 utterances (15,252 min), *i.e.*, 3,200 $\times$ (8 noises $\times$ 5 levels + 1 clean), the validation set contains 23,616 utterances (2,542 min), and the test set contains 57,584 utterances (6,344 min), *i.e.*, 944 $\times$ (12 noises $\times$ 5 levels + 1 clean).

TABLE IV
TYPES OF ADDITIVE NOISE USED IN THE EXPERIMENTS. THE NAMES REFER TO THE NOISES IN THE NOISEX-92 DATASET.

Type (NOISEX-92)	Training/Validation	Test	Condition
(Clean) Babble Destroyerops F16 Factory2 Leopard M109 Machinegun White	Yes	Yes	**Known**
Destroyerengine Factory1 Pink Volvo	No	Yes	**Unknown**

C. Training and Test Settings

The speech signals in the datasets are divided into 25-ms frames at 5 ms intervals. The first 400 and the last 200 frames in each utterance are removed to reduce non-speech frames. For REG(wav2sin), a sequence of seven consecutive frames, *i.e.*, $\mathbf{u}_{i-3}$ to $\mathbf{u}_{i+3}$, is input into the RNN to estimate $\hat{f0}_i$. For a frequency-domain analysis of REG(sp2f0), the STFT is applied with 1024-point DFT to obtain the power spectral density (PSD) of the time-frequency domain and then a sequence of seven consecutive frames comprising the first 513 bins in the frequency domain, *i.e.*, $0 \le \omega \le \pi$, at each frame are used to obtain the estimate $\hat{f0}_i$. The procedures for feature extraction and F0 quantization for CLS(sp2f0), CLS(wav2f0) and CREPE follow [20], [22] and [13], respectively, but some parameters are modified to match our test conditions and datasets by preliminary experiments as follows.

- The waveform input of CLS(wav2f0) and CREPE comprises 400 samples
- The input of CLS(sp2f0) consists of 3591 DFT bins (513 bins $\times$ 7 frames)
- Each hidden layer of CLS(sp2f0) and CLS(wav2f0) consists of 1024 units
- The output layer of CREPE is modified with the same quantization as CLS(wav2f0) because the original settings in CREPE are for music signals.

The hyperparameters of the proposed methods are determined in accordance with our previous works [15], [24], in which the networks are determined considering the balance between performance and computational complexity. The settings for REG(sp2f0) and REG(wav2sin) are summarized in Tables V and VI, respectively, and the datasets for the

TABLE V
NETWORK SETTINGS FOR REG(SP2F0)

Parameter	Input Layer	Hidden Layers	Output Layer
# of Layers	1	3	1
# of Units	3591	1024	1
Activation	-	$\tanh$	Identity
Memory Cells	LSTM		
Minibatch	200		
Context	q = 3		

TABLE VI
RNN SETTINGS FOR REG(WAV2SIN)

Parameter	Input Layer	Hidden Layers	Output Layer
# of Layers	1	3	1
# of Units	400	1024	400
Activation	-	$\tanh$	Identity
Memory Cells	LSTM		
Minibatch	200		
Context	q = 3		

experiment are same as in Table III. The number of hidden layers (LSTM cells) is set equal to three with 1024 units each that are activated by the $\tanh$-function. The minibatch size is set to 200 frames, and random unit dropout (25 %) and batch normalization [47] are applied during training.

D. Performance Metrics

The performance of F0 tracking is evaluated using the three standard metrics: the *gross pitch error* (GPE) rate, the *fine pitch error* (FPE) and the *voicing error rate* (VER) [31]. GPE frames are voiced frames in which the error between the estimate of the pitch period ($1/\hat{f}0$) and the ground truth ($1/f0$) is greater than the period corresponding to 10 samples, *i.e.*, 0.625 ms. Formally,

$$\text{GPE rate} = \frac{N_{\text{GPE}}}{N_v}, \tag{39}$$

where N_{GPE} and N_v denote the number of GPE frames and voiced frames per utterance, respectively. The FPE frames, in turn, are voiced frames after excluding the GPE frames. The mean FPE (μ_{FPE}) represents the *bias* in the F0 estimation, while the standard deviation of the FPE (σ_{FPE}) measures the *accuracy* [31]. Formally,

$$\mu_{\text{FPE}} = \frac{1}{N_{\text{FPE}}} \sum_{i=1}^{N_{\text{FPE}}} \epsilon_i \tag{40}$$

$$\sigma_{\text{FPE}} = \sqrt{\frac{1}{N_{\text{FPE}}} \sum_{i=1}^{N_{\text{FPE}}} (\epsilon_i - \mu_{\text{FPE}})^2} \tag{41}$$

$$\epsilon_i = \left| \hat{f}_0^i - f_0^i \right|, \tag{42}$$

where $\hat{f}_0^i$ and f_0^i denote the estimate and the ground truth of F0, respectively, at the i-th frame in the FPE frames, while N_{FPE} is the number of FPE frames.

Finally, the VER is the ratio of voicing error frames to the total number of frames:

$$\text{VER} = \frac{N_{\text{VE}}}{N} \tag{43}$$

where N_{VE} and N denote the number of voicing error frames and total frames, respectively.

V. Experimental Results

A. Gross and Fine Pitch Error

Panels (a) and (b) of Fig. 12 illustrate the GPE rates for each method at different SNRs in the multi-noise conditions for the known and unknown noises, respectively.

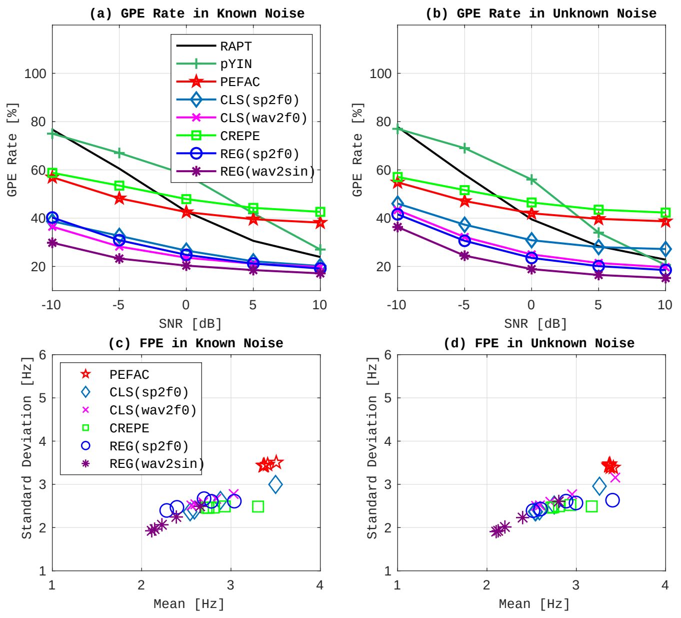

Fig. 12. Estimation accuracy of each F0 tracker at different SNRs. (a) The gross pitch error (GPE) rates for the known noises, (b) the GPE rates for the unknown noises, (c) the scatter plot of the mean and standard deviation of the fine pitch errors (FPE), μ_{FPE} and σ_{FPE}, respectively, for the known noises, and (d) the FPE performance for the unknown noises.

The GPE rate has a decreasing trend at higher SNRs across all the methods, and the supervised machine learning approaches, except CREPE, show more difficulty with unknown noises than known noises in terms of the GPE rate, whereas the filtering-based approaches are less affected by the difference between known and unknown noise. Nevertheless, the machine learning approaches (except for CREPE) mostly outperform the filtering-based methods by a wide margin.

REG(wav2sin) outperforms the other methods in terms of the GPE rate. For instance, it yields a GPE rate of 30 % for known noises and 37 % for unknown noises at an SNR of -10 dB. REG(sp2f0) also yields a lower GPE rate than the previous reference methods but is outperformed by REG(wav2sin) by approximately 6 percentage points or more at 0 dB and below under both noise conditions.

GPE frames correspond to failure in F0 estimation for voiced frames [31]. In that sense, F0 estimation with pYIN at SNRs below 5 dB, RAPT at SNRs less than 0 dB and CREPE and PEFAC at SNRs of -5 dB and below are likely to have unreliable frames accounting for more than 50 % of the voiced frames. Conversely, REG(wav2sin) retains its estimation error for voiced frames at approximately 35 %, even at -10 dB for unknown noises, while the corresponding errors for REG(sp2f0) and CLS(wav2f0) are at approximately 40 %. These observations suggest a substantial advantage of our proposal, in particular, the waveform-to-sinusoid approach.

The FPE error analysis is depicted in panels (c) and (d) of Fig. 12 as scatter plots of μ_{FPE} and σ_{FPE} corresponding to the tested SNRs, *i.e.* -10, -5, 0, +5 and +10 dB. pYIN and RAPT are excluded from this analysis because neither method provided sufficiently many FPE frames under noisy conditions.

REG(wav2sin) performs the best in terms of both the bias (mean) and accuracy (standard deviation). Although PEFAC displays strong noise robustness in both metrics, REG(wav2sin) outperforms it by approximately 35 % in relative percentage on average under both noise conditions. As with the GPE rate, REG(wav2sin) outperforms REG(sp2f0) by more than 15 %, which is at the same level of performance as both CLS methods and CREPE in terms of μ_{FPE} and σ_{FPE}.

Overall, the regression tasks are more difficult than classifying speech frames or spectral features into quantized frequency states. However, RNN regression may capture temporal dynamics better by optimizing the recurrent weights than the fully connected DNNs in the CLS methods; even if the fully connected DNNs augment their input with some preceding and successive frames, this augmentation induces many poorly correlated connections to the network structure. Consequently, the RNNs have achieved regression tasks that outperform the classification approaches.

B. Influence of the Subsequence Length

In the proposed methods, REG(sp2f0) and REG(wav2sin), we use a subsequence centering around the i-th frame with a context length of $2q+1$. In the preceding experiments, we used $q = 3$. Here, we are interested in studying how the choice of q impacts performance. Table VII summarizes the GPE rate, μ_{FPE} and σ_{FPE} with different subsequence lengths under the clean and noisy conditions. The noisy condition is represented by unknown noises at an SNR of 0 dB. The GPE rate varies across different subsequence lengths and is minimized at approximately $q = 3$, while the FPE does not show substantial variation within the range $q \in [2..6]$.

TABLE VII
Performance with different subsequence lengths q. Here, *noisy* refers to unknown noise at an SNR of 0 dB.

			Subsequence length parameter, q				
			2	3	4	5	6
GPE Rate [%]	sp2f0	Clean	13.55	**12.19**	13.12	13.06	16.23
		Noisy	24.29	23.53	**23.44**	24.26	27.41
	wav2sin	Clean	12.49	**11.88**	12.04	12.53	13.21
		Noisy	19.26	**18.89**	18.97	19.41	22.32
FPE μ [Hz]	sp2f0	Clean	2.4	**2.3**	**2.3**	2.4	**2.3**
		Noisy	2.8	2.8	2.8	2.8	2.8
	wav2sin	Clean	1.9	**1.8**	**1.8**	1.9	2.0
		Noisy	2.3	2.3	2.3	2.3	2.3
FPE σ [Hz]	sp2f0	Clean	2.3	**2.2**	**2.2**	**2.2**	**2.2**
		Noisy	2.6	2.6	2.6	2.6	2.6
	wav2sin	Clean	**1.9**	**1.9**	**1.9**	2.0	2.1
		Noisy	**2.0**	**2.0**	**2.0**	**2.0**	2.1

C. Voicing Detection

To analyze voicing errors, we apply the bi-Gaussian voicing detector to the proposed regression methods. Because neither RAPT nor pYIN provides sufficient noise robustness (Fig. 12), we decided to evaluate PEFAC as the representative of the signal processing methods. We use the native voicing detector of PEFAC. This detector estimates the posterior probability of voiced and unvoiced classes and declares a frame voiced if the former is larger [14], which coincides with the decision rule used in our voicing detectors in (20). The classification approaches do not require any additional processes for voicing determination, as discussed earlier. We use CLS(sp2f0) and CLS(wav2f0) as representatives of the classification methods because their robustness to noise substantially exceeds that of CREPE.

Fig. 13 represents the voicing error rates of each method at different SNRs. REG(sp2f0) always outperforms the other

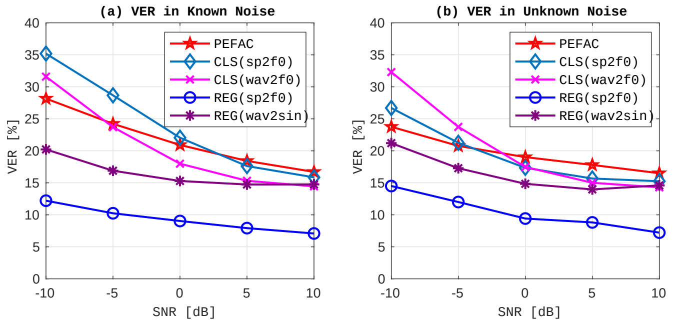

Fig. 13. Voicing error rates at different SNRs under (a) the known noise condition and (b) the unknown noise condition.

models, sustaining a voicing error rate between 7 % and 15 % across all the conditions. REG(wav2sin) follows, but the gap in the performance to REG(sp2f0) is always more than 5 % absolute. The main difference between the two voicing detectors are in the employed features — $\hat{f0}_i$ for REG(sp2f0) and σ_{yi}^2 and r_{yi} for REG(wav2sin). CLS(sp2f0) and CLS(wav2f0) yield lower VERs than PEFAC, between 15 % and 22 %, at SNRs higher than 0 dB, though their VERs exceed the VER of PEFAC for SNRs below 0 dB, despite lower GPE and FPE rates (Fig. 12).

Fig. 14 illustrates typical distributions of voiced and unvoiced classes in our bi-Gaussian voicing detector used in REG(sp2f0). The voiced and unvoiced classes are clearly distinct for clean data, and separation between the classes is higher for the female speaker. In the noise-contaminated case, the distribution of the unvoiced class is shifted towards the voiced class as noise dominates the unvoiced frames. The classes, however, are still separated.

Fig. 15 illustrates the bivariate Gaussian distributions in REG(wav2sin). For improved visualisation, x-axes display $\sqrt{2}\sigma_{yi}$ instead of σ_{yi}^2. Voiced and unvoiced classes can be separated at the contour of equal posterior probabilities, $P(\mathbf{a}_i|\mathcal{H}_0) = P(\mathbf{a}_i|\mathcal{H}_1)$, visualized with the thick magenta line. Due to class-dependent covariances, the resulting decision surfaces are quadratic. Variance of the features is generally much lower within the unvoiced class. Compared with REG(sp2f0),

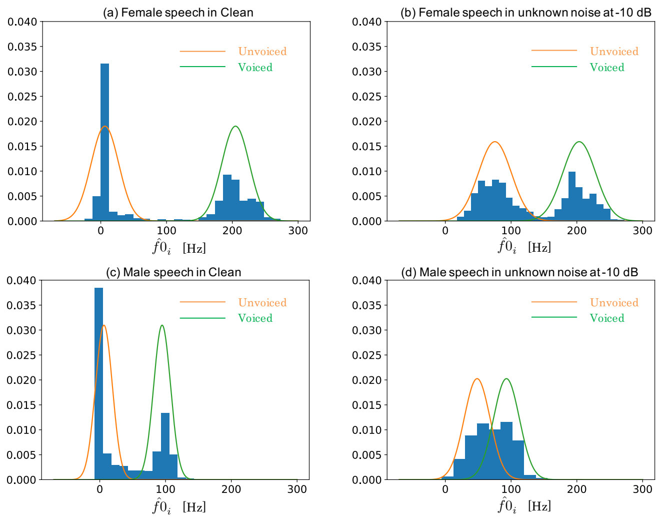

Fig. 14. (a) and (b) illustrate the bivariate Gaussian distributions of a female utterance under the clean and noisy conditions respectively. (c) and (d) depict the distributions of a male utterance under the clean and noisy conditions respectively. The noisy condition is represented by Factory1 at SNR of -10 dB.

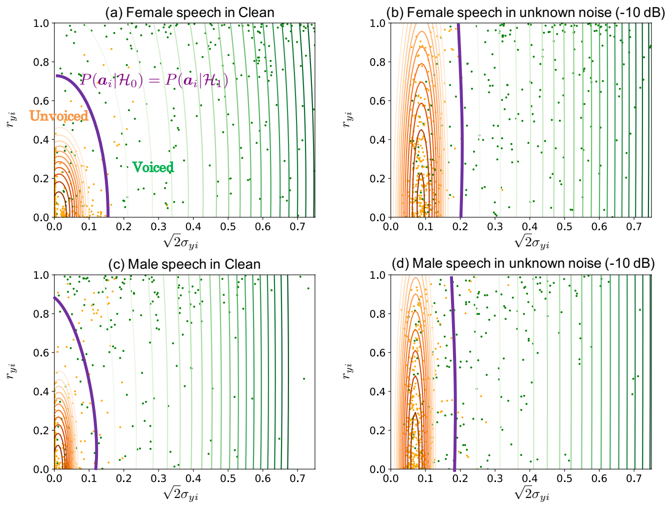

Fig. 15. (a) and (b) illustrate the bivariate Gaussian distributions of a female utterance under the clean and noisy conditions respectively. (c) and (d) depict the distributions of a male utterance under the clean and noisy conditions respectively. The noisy condition is represented by Factory1 at SNR of -10 dB.

the probability densities seem to depend less on speakers, as the used features lack direct dependency on $\hat{f0}_i$ values. In the noise-contaminated case, the distribution of the unvoiced class expands; as the noise components tend to be mapped onto a sinusoid as voiced speech, this reduces voicing detection accuracy in noisy conditions.

The preceding results demonstrate that the voicing determination problem in our proposed regression-based F0 trackers can be effectively addressed through the bi-Gaussian approach: the voicing error rates in both methods are lower than those of the reference methods. We expect that further improvement is possible through alternative robust features (or alternative distributions), which is left for future work.

D. F0 Contours

Fig. 16 illustrates F0 contours of the spoken word "DARK" estimated by CLS(wav2f0), CREPE, REG(sp2f0) and REG(wav2sin) under clean conditions, and they are compared with the ground truth (REF). Panels (a) and (b) show the F0 contours spoken by a female speaker and a male speaker, respectively, and each contour is shifted for better visualization. The utterances of these two speakers are not included in the training set, *i.e.*, unknown speakers. The figures demonstrate the advantage of our proposed regression approaches (REGs) over the existing classification approaches (CLS and CREPE) in terms of the naturalness of the F0 contours. This advantage also reveals the potential of our proposal to track the prosody of different speakers in a speaker-independent manner.

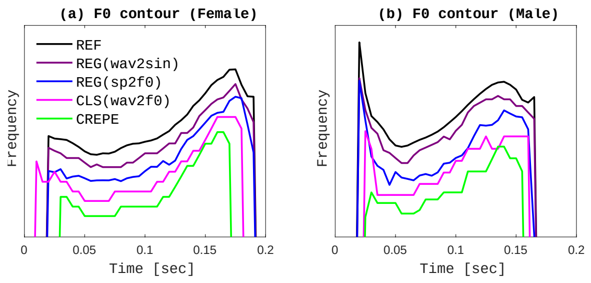

Fig. 16. F0 contours of word "DARK" spoken by (a) an unknown female speaker and (b) an unknown male speaker. Each contour is shifted for better visualization.

VI. Conclusion

We addressed the problem of F0 tracking with RNN-based regression techniques to obtain accurate and natural F0 contours with improved noise robustness. The proposed regression approaches, especially the waveform-to-sinusoid regression approach, demonstrate considerable improvement over the existing state-of-the-art F0 trackers. Compared to PEFAC, one of the most robust autocorrelation-based F0 trackers, the waveform-to-sinusoid regression method, REG(wav2sin), yielded a relative improvement exceeding 35 % in both the gross pitch error (GPE) rate and the fine pitch error (FPE) at SNRs between -10 dB and +10 dB under both known and unknown noise conditions. Furthermore, our proposed waveform-to-sinusoid regression method outperformed the latest DNN and CNN-based F0 trackers in terms of relative improvement in both the GPE rate and the FPE by more than 15 % over the preceding SNR range.

The comparison of the estimated F0 contours of clean speech also demonstrates an advantage of our proposal over other DNN- and CNN-based approaches in producing more natural F0 trajectories. While the present work focused solely on the F0 tracking problem itself, our future plan involves integrating the proposed method in a downstream application such as voice conversion or prosody-based speaker recognition.

Acknowledgment

This work was supported in part by the Academy of Finland (Proj. No. 309629). The authors wish to acknowledge CSC - IT Centre for Science, Finland, for the computational resources.

References

[1] X. Wang, S. Takaki, and J. Yamagishi, "An RNN-based quantized F0 model with multi-tier feedback links for text-to-speech synthesis," *Proceedings of INTERSPEECH*, pp. 20–24, August 2017.
[2] S. H. Mohammadi and A. Kain, "An overview of voice conversion systems," *Speech Communication*, vol. 88, pp. 65–82, 2017.
[3] E. Godoy, J. R. Williamson, and T. F. Quatieri, "Canonical correlation analysis and prediction of perceived rhythmic prominences and pitch tones in speech," *Proceedings of INTERSPEECH*, pp. 3206–3210, August 2017.
[4] V. Rajendran, A. A. Kandhadai, and V. Krishnan, "Systems, methods, and apparatus for signal encoding using pitch-regularizing and non-pitch-regularizing coding," *US Patent 9,653,088*, 2017.
[5] A. Kato and B. Milner, "Using hidden Markov models for speech enhancement," *Proceedings of INTERSPEECH*, pp. 2695–2699, 2014.
[6] ——, "HMM-based speech enhancement using sub-word models and noise adaptation," *Proceedings of INTERSPEECH*, pp. 3748–3752, September 2016.
[7] P. A. Torres-Carrasquillo, F. Richardson, S. Nercessian, D. Sturim, W. Campbell, Y. Gwon, S. Vattam, N. Dehak, H. Mallidi, P. S. Nidadavolu *et al.*, "The MIT-LL, JHU and LRDE NIST 2016 speaker recognition evaluation system," *Proceedings of INTERSPEECH*, pp. 1333–1337, August 2017.
[8] D. Nandi, D. Pati, and K. S. Rao, "Parametric representation of excitation source information for language identification," *Computer Speech and Language*, vol. 41, pp. 88–115, January 2017.
[9] D. Talkin, "A robulst algorithm for pitch tracking (RAPT)," *Speech coding and synthesis*, pp. 495–518, 1995.
[10] A. De Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, April 2002.
[11] M. Mauch and S. Dixon, "pYIN: A fundamental frequency estimator using probabilistic threshold distributions," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 659–663, May 2014.
[12] D. Wang, P. C. Loizou, and J. H. Hansen, "F0 estimation in noisy speech based on long-term harmonic feature analysis combined with neural network classification," *Proceedings of INTERSPEECH*, pp. 2258–2262, September 2014.
[13] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "CREPE: A convolutional representation for pitch estimation," *arXiv preprint arXiv:1802.06182*, February 2018.
[14] S. Gonzalez and M. Brookes, "PEFAC - A pitch estimation algorithm robust to high levels of noise," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 2, pp. 518–530, February 2014.
[15] A. Kato and T. Kinnunen, "A regression model of recurrent deep neural networks for noise robust estimation of the fundamental frequency contour of speech," *Proceedings of Odyssey, The Speaker and Language Recognition Workshop*, pp. 275–282, June 2018.
[16] B. Milner and X. Shao, "Prediction of fundamental frequency and voicing from Mel-frequency cepstral coefficients for unconstrained speech reconstruction," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 24–33, 2007.
[17] Z. Jin and D. Wang, "HMM-based multipitch tracking for noisy and reverberant speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 5, pp. 1091–1102, July 2011.
[18] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura, "Speech synthesis based on hidden Markov models," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1234–1252, May 2013.
[19] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, November 2012.
[20] K. Han and D. Wang, "Neural network based pitch tracking in very noisy speech," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 22, no. 12, pp. 2158–2168, December 2014.

[21] D. Wang, C. Yu, and J. H. L. Hansen, "Robust harmonic features for classification-based pitch estimation," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 25, no. 5, pp. 952–964, May 2017.

[22] P. Verma and R. W. Schafer, "Frequency estimation from waveforms using multi-layered neural networks," *Proceedings of INTERSPEECH*, pp. 2165–2169, September 2016.

[23] X. Wang, S. Takaki, and J. Yamagishi, "Autoregressive neural f0 model for statistical parametric speech synthesis," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 26, no. 8, pp. 1406–1419, August 2018.

[24] A. Kato and T. Kinnunen, "Waveform to single sinusoid regression to estimate the F0 contour from noisy speech using recurrent deep neural networks," *Proceedings of INTERSPEECH*, pp. 327–331, September 2018.

[25] A. Gibiansky, S. Arik, G. Diamos, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, "Deep voice 2: Multi-speaker neural text-to-speech," *Advances in neural information processing systems*, pp. 2962–2970, 2017.

[26] K. Yu and S. Young, "Continuous F0 modeling for HMM based statistical parametric speech synthesis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 5, pp. 1071–1079, 2010.

[27] I. Magrin-Chagnolleau, G. Gravier, and R. Blouet, "Overview of the 2000-2001 ELISA consortium research activities," *Proceedings of Odyssey, The Speaker and Language Recognition Workshop*, 2001.

[28] A. Sholokhov, M. Sahidullah, and T. Kinnunen, "Semi-supervised speech activity detection with an application to automatic speaker verification," *Computer Speech and Language*, vol. 47, pp. 132–156, 2018.

[29] G. D. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[30] M. Wu, D. Wang, and G. J. Brown, "A multipitch tracking algorithm for noisy speech," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 3, pp. 229–241, 2003.

[31] L. Rabiner, M. Cheng, A. Rosenberg, and C. McGonegal, "A comparative performance study of several pitch detection algorithms," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 24, no. 5, pp. 399–418, October 1976.

[32] Y. Bengio, P. Simard, P. Frasconi *et al.*, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[34] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv:1412.3555*, 2014.

[35] F. Weninger, H. Erdogan, S. Watanabe, E. Vincent, J. Le Roux, J. R. Hershey, and B. Schuller, "Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR," *Proceedings of International Conference on Latent Variable Analysis and Signal Separation*, pp. 91–99, August 2015.

[36] F. A. Gers, D. Eck, and J. Schmidhuber, "Applying lstm to time series predictable through time-window approaches," *Neural Nets WIRN Vietri-01*, pp. 193–200, 2002.

[37] A. Graves, S. Fernández, and J. Schmidhuber, "Bidirectional LSTM networks for improved phoneme classification and recognition," *Proceedings of International Conference on Artificial Neural Networks*, no. 799-804, 2005.

[38] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4945–4949, 2016.

[39] L. Sun, S. Kang, K. Li, and H. Meng, "Voice conversion using deep bidirectional long short-term memory based recurrent neural networks," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4869–4873, 2015.

[40] Y. Fan, Y. Qian, F.-L. Xie, and F. K. Soong, "TTS synthesis with bidirectional LSTM based recurrent neural networks," *Proceedings of INTERSPEECH*, 2014.

[41] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," 1985.

[42] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification and scene analysis*. Wiley New York, 1973, vol. 3.

[43] D. A. Van Leeuwen and N. Brümmer, "An introduction to application-independent evaluation of speaker recognition systems," *Speaker classification I*, pp. 330–353, 2007.

[44] S. Cumani and P. Laface, "Tied normal variance–mean mixtures for linear score calibration," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6121–6125, 2019.

[45] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[46] G. Pirker, M. Wohlmayr, S. Petrik, and F. Pernkopf, "A pitch tracking corpus with evaluation on multipitch tracking scenario," *Proceedings of INTERSPEECH*, pp. 1509–1512, 2011.

[47] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *Proceedings of International Conference on Machine Learning*, vol. 37, pp. 448–456, July 2015.

Akihiro Kato received the PhD degree from the University of East Anglia, Norwich, UK, in 2017, and the BEng degree from Chuo University, Tokyo, Japan, in 2001. He carried out research at the University of Eastern Finland from 2017 to 2018. Currently, he is a research scientist at Ricoh Institute of Technology of Ricoh Company, Ltd. His research interest is machine learning approaches to speech processing, including automatic speech recognition, speech conversion, speech synthesis and speech enhancement.

Tomi H. Kinnunen is an Associate Professor at the University of Eastern Finland. He received the Ph.D. degree in computer science from the University of Joensuu in 2005. From 2005 to 2007, he was an Associate Scientist at the Institute for Infocomm Research (I2R) in Singapore. Since 2007, he has been with UEF. From 2010-2012, he was funded by a postdoctoral grant from the Academy of Finland focusing on speaker recognition. He has been a PI or co-PI in three other large Academy of Finland-funded projects on speaker recognition and voice antispoofing and a partner in the H2020-funded OCTAVE project focusing on voice biometrics for physical and logical access control. He chaired the *Odyssey* workshop in 2014. From 2015 to 2018, he served as an Associate Editor for IEEE/ACM Trans. on Audio, Speech and Language Processing and from 2016 to 2018 as a Subject Editor in *Speech Communication*. Between 2015 and 2016, he visited the National Institute of Informatics (NII), Japan, for 6 months under a mobility grant from the Academy of Finland, with a focus on voice conversion and spoofing. Since 2017, he has been Associate Professor at UEF, where he leads the Computational Speech Group. He is known as one of the cofounders of the ASVspoof Challenge, a nonprofit initiative that seeks to evaluate and improve the security of voice biometric solutions under spoofing attacks.