Tonal Contour Generation for Isarn Speech Synthesis Using Deep Learning and Sampling-Based $F_0$ Representation

Pongsathon Janyoi † and Pusadee Seresangtakul * †
Natural Language and Speech Processing Laboratory (NLSP), Department of Computer Science, Khon Kaen University, Khon Kaen 40002, Thailand; pongsathon.janyoi@kkumail.com
* Correspondence: pusadee@kku.ac.th
† These authors contributed equally to this work.

Received: 17 August 2020; Accepted: 10 September 2020; Published: 13 September 2020

Abstract: The modeling of fundamental frequency ($F_0$) in speech synthesis is a critical factor affecting the intelligibility and naturalness of synthesized speech. In this paper, we focus on improving the modeling of $F_0$ for Isarn speech synthesis. We propose the $F_0$ model for this based on a recurrent neural network (RNN). Sampled values of $F_0$ are used at the syllable level of continuous Isarn speech combined with their dynamic features to represent supra-segmental properties of the $F_0$ contour. Different architectures of the deep RNNs and different combinations of linguistic features are analyzed to obtain conditions for the best performance. To assess the proposed method, we compared it with several RNN-based baselines. The results of objective and subjective tests indicate that the proposed model significantly outperformed the baseline RNN model that predicts values of $F_0$ at the frame level, and the baseline RNN model that represents the $F_0$ contours of syllables by using discrete cosine transform.

Keywords: tone; fundamental frequency; recurrent neural networks; Isarn dialect; speech synthesis

1. Introduction

The fundamental frequency ($F_0$) contour plays an important role in speech synthesis systems. The $F_0$ contour controls the intelligibility and naturalness of synthetic speech. In the speech synthesis of a tonal language, tone is correlated with $F_0$. Tone correctness is crucial because words with different tones convey different meanings even if their phonemes are similar [1]. Thus, it is necessary to generate an appropriate tone contour for tonal languages.

Several studies have proposed speech synthesis for tonal languages [2–4] and a few have developed speech synthesis for the Isarn language [5], a dialect of Thai. Isarn is classified as a low-resource language. In our previous work [5], the hidden Markov model (HMM)-based speech synthesis was proposed for Isarn which can generate synthetic speech with an acceptable level of naturalness. However, the generation of inappropriate $F_0$ contour still degrades the naturalness of synthetic speech because considering the values of $F_0$ frame by frame is insufficient to model the suprasegmental features of the $F_0$ contour [6,7].

In the context of tonal languages, several studies have attempted to improve the modeling of $F_0$ for HMM-based speech synthesis. For example, a simple tone-separated tree structure with contextual tonal information was proposed to improve the correctness of the tone of HMM-based speech synthesis in Thai [8]. Multi-layer $F_0$ models for HMM-based speech synthesis have been proposed as well [6,7,9] by using models of $F_0$ to represent its patterns for different prosodic layers. These proposals can improve tone correctness and the naturalness of synthetic speech. However, performance is limited by decision tree-based clustering [10].
In the last few years, several types of neural network-based methods of speech synthesis have been proposed to overcome the limitations of HMM-based speech synthesis. Deep neural networks (DNNs) [10] have been used in place of decision tree-based clustering. However, DNN ignores the sequential nature of speech because it assumes that each frame is sampled independently. To capture long-term dependencies, the recurrent neural network (RNN) with long-short term memory (LSTM) has also been applied as an acoustic model [11–13].

End-to-end speech synthesis using advanced neural network architectures has also been proposed, such as in Tacotron [14] and Deep Voice [15]. These techniques directly convert raw text into speech waveforms and generate speech that sounds more natural than both parametric-based and concatenative approaches. However, they require a large set of speech and text pairs for training, which is time consuming and expensive to collect [16]. These models are thus challenging to apply to low-resource languages. Thus, this study focuses on parametric-based speech synthesis approach.

As described in [17,18], the shape of the $F_0$ contour of a syllable can deviate due to several factors, such as tone co-articulation, stress, and declination effects. Thus, the contour should be modeled at the syllable level rather than the frame level. Several models are available for representing the $F_0$ contour, such as the tilt model [19,20], Fujizaki’s model [21], pitch target approximation models [22], and the discrete cosine transform (DCT) [23,24]. These models substantially improve performance in terms of the generation of $F_0$. Nevertheless, some $F_0$ representation models require manual annotation to correct (e.g., the Fujizaki and pitch target approximation models) [25,26].

Although the RNN can capture suprasegmental characters of the $F_0$ contour, its internal structure of connections considers $F_0$ at the frame or segment level only. In particular for tonal languages, tone contour and its deviation are also characterized at higher levels (e.g., syllable and word) [27–29].

In this paper, we propose a tone modeling technique based on the RNN for speech synthesis in Isarn. We first propose a sampling-based method of representation of $F_0$ that can capture rich information concerning the tone contours within a syllable for Isarn. Syllable-level features are then modeled by the RNN to learn the suprasegmental characters of $F_0$ at higher prosodic levels. We also explore several architectures of RNN models and training strategies to generate tone contours. In terms of the linguistic level used for modeling and the representation of $F_0$, we compare the proposed model with a frame-based model and other established $F_0$ transforms, such as the DCT, using objective and perceptual evaluations.

The remainder of the paper is organized as follows: Section 2 briefly introduces the Isarn language and its tone, challenges posed by tone modeling, and past work in speech synthesis for Isarn. Section 3 describes the proposed $F_0$ model, and Section 4 describes the experiments and results. Our conclusions and recommendations for future research are presented in the final section.

2. Background and Related Studies

2.1. Isarn Language and Tone

Isarn is a tonal language spoken in Northeastern Thailand. Locals speak Isarn in different dialects depending on the region. Some Isarn dialects have five tones but others have six [30]. In this research, we focus on the dialect spoken in Central-Northeast Thailand (covering Khon Kaen, Kalasin, Udonthani, and Mahasarakham provinces). There are six tones in the central Isarn dialect: Mid (M), low (L), mid-falling (MF), high falling (HF), high (H), and rising (R). The following examples show a word consisting of the same phonemes in different tones: M /kʰaː/ (“cost”), L /kʰàː/ (“kill”), MF /kʰâː/ (“trade”), HF /kʰáː/ (“stick”), H /kʰáː/ (“galangal”), and R /kʰáː/ (“leg”). Figure 1 shows the typical pattern of each tone analyzed using the speech of a native male speaker.
In the past, the Isarn was written using the Isarn Dharma script. The difficulty of representing the Isarn by the Isarn Dharma script is its written style which lacks tone markers to identify the tones [31]. Nowadays, people mostly use Thai script, which is the official script of Thailand, to write the Isarn language. Although Thai script has tone markers, there is no formal standard rule for written Isarn language [32]. The written depends on the personal style. The same grapheme words may pronounce in different sounds with different meanings depending on surrounding words. This leads to ambiguity problems in pronunciation and text processing [33]. For example, the Isarn word “ยำม” can be pronounced as /jaːm/ (“visit”) in the sentence “ข้อไปตามเข้าบ้าน” /kʰoŋjaːm câw já bâːm/ (means “I visit you at home” in English) or “/nâm/” (“time”) in the sentence “ยำมได้ขึ้นโรงเรียน” /nâm dâːj câw sî pâj lûŋ liâːn/ (means “What time will you go to school?” in English). In this study, the conversion of text into a linguistic specification is achieved using the front-end module [33] and corrected manually using audio recordings for reference.

2.2. Challenges of Tone Modeling

In tonal languages, tone behavior is complex in continuous speech even though there is only a finite number of tones. Several studies have been carried out on deviations in tone such languages as Thai and Mandarin [18,34–36], but Isarn has not yet been thus considered. We examined these studies for the extracted $F_0$ contours from Isarn to identify factors affecting deviations in tone contours in continuous speech. Based on past studies, many factors affect deviations in tonal contours in continuous speech, such as tone co-articulation, stress, and declination. Figure 2 shows a comparison of the $F_0$ contour of a sentence in Isarn pronounced in isolation and continuously. As is shown, the $F_0$ contour of continuous speech deviated due to these factors.

Tone co-articulation is a phenomenon whereby the shape of the $F_0$ contour of a given syllable is affected by $F_0$ contours of adjacent syllables because the articulatory organs cannot respond rapidly enough to preserve the shape of the $F_0$ contours of the uttered syllables. For example, in Figure 2b, the $F_0$ contour of syllable /mân/ is assimilated into syllable /mâː/. Note that shapes of their $F_0$ contours differ from the patterns of their pronunciations in isolation as shown in Figure 2a. The $F_0$ contours of stress syllables differ from those of unstressed syllables and more closely approximate a stable pattern than them—for instance, the $F_0$ contour of the syllable /mân/ (without stress) and that of the syllable /dâː/ (with stress). The declination effect refers to the downward trend of the $F_0$ level to conform to the pattern of intonation of a larger prosodic level, such as that of a phrase or a sentence.
2.3. Past Work on Isarn Speech Synthesis

HMM-based speech synthesis for Isarn was developed in [5]. The waveform of the speech units was not used directly, instead linguistic and acoustic features were extracted from a speech corpus. In the training stage, the waveform of speech is converted into acoustic features, including $F_0$ and spectral features. The components of the acoustic parameters depend on the speech vocoder. Transcription is converted into linguistic features used as input by a text analysis module. The acoustic models are then trained using the extracted acoustic features and linguistic features. A duration model is also trained to determine phone duration. In the synthesis stage, linguistic contextual features are extracted from the input text and fed to the duration model. A sequence of speech parameters is then generated using the trained acoustic model and the information obtained concerning the phone duration. Finally, the waveform of speech is synthesized through the speech vocoder.

In HMM-based speech synthesis for Isarn, the tonal syllable is modeled by using two or three contextual phone models, including the initial phone model, vowel phone model, and final phone model (optional). The six tones in Isarn are represented in terms of tone–context features.

However, the HMM-based speech synthesis for Isarn still generates unnatural speech related to the generation of $F_0$. Therefore, we examined past studies to identify a method to improve the performance of Isarn speech synthesis. We found that RNN-based speech synthesis had achieved the best performance for other languages [11,12,37,38]. We also preliminarily implemented RNN-based speech synthesis for Isarn, and observed the synthetic speech. RNN-based speech synthesis often generated unnatural speech due to the generation of inappropriate $F_0$ contours. Therefore, we attempted to improve the $F_0$ model for Isarn speech synthesis.

At a linguistic level, the proposed model does not consider temporal dependencies across frames, but across syllables. In tonal languages, the tone is indicated by the $F_0$ contour at the syllable level [1]. The deviation in the $F_0$ contour also occurs across syllables. From the perspective of

---

**Figure 2.** Comparison of $F_0$ contours of the Isarn utterance (“Please call him to meet the teacher.” in English translation) pronounced by the same speaker in (a) isolation and (b) continuous speech.
feature representation, we propose a sampling-based approach that represents the $F_0$ contour of the syllable by using sampled $F_0$ values and their dynamic features. We expect that the sampling-based method can provide rich information for modeling suprasegmental features of the $F_0$ contour.

This model consists of two steps. First, the $F_0$ contour of the syllables is represented by the sampling-based method. Second, linguistic features are mapped into the extracted parameters by using the RNN. The development of the proposed model can be divided into two parts: $F_0$ contour modeling and synthesis, as shown in Figure 3. Details of each part are described below.

**Figure 3.** An overview of the proposed $F_0$ model.

### 3. Proposed Method

#### 3.1. $F_0$ Contour Modeling

We construct the $F_0$ model based on the RNN. This part consists of three processes: Linguistic feature extraction, sampling-based $F_0$ representation, and model training. The details of each process are provided below.

#### 3.1.1. Linguistic Feature Extraction

In this section, we formally describe the representation of input features for modeling the $F_0$ contour. We extracted linguistic features by using a question set and context-dependent labels of the HMM-based Isarn speech synthesis [5]. The linguistic features considered are based on multi-prosodic layers, such as syllable, word, intermediate phrase, intonation phrase, and utterance as listed in Table 1. Linguistic features can be divided into four parts: Tone, phone identity, position, and features of duration.

Tonal features are important for predicting the $F_0$ contour of syllables for Isarn speech synthesis, as investigated in our previous work [5]. Tonal features can help infer the rough curve of the contours of a tone. However, other contextual features are needed to perfectly model the $F_0$ contour because using tonal features alone cannot be used to deal with complex variations in the $F_0$ contour.

Features of phone identity are combined to represent a syllable. Each syllable is represented by a one-hot vector encoding the phone identity and phone categories of the initial consonant, vowel, and final consonant. In Isarn, the final consonant is optional. We add a flag here to indicate the given syllable that is not the final consonant.

Positional features relate the number and positions of syllables in the higher prosodic layers. When examining examples of utterances in the Isarn speech corpus, we found that the $F_0$ contour is also
related to stress information, however annotation is time consuming and expensive. Stress syllables typically have a long duration and appear at the end of an utterance or intonation [34]. We thus included sectional features that consider the given position of the syllable at a higher prosodic level and duration-related features. The effect of each feature set is investigated in Section 4.4.2.

Table 1. Linguistic features for training the proposed model.

| Feature Category   | Description                                                                                                                                   |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| Tone features      | Tone identities of previous/current/next syllable.                                                                                           |
| Duration feature   | Duration of current syllable. Duration of initial consonant/vowel/final consonant of current syllable.                                      |
| Phoneme features   | Phoneme identities of initial consonant/vowel/final consonant of current syllable. Phoneme categories of initial consonant/vowel/final consonant of current syllable. |
| Positional features| Position of the current syllable in the current word. Number of syllables in the current word. Position of the current syllable in the current phrase. Number of syllables in the current phrase. Position of the current syllable in the current intermediate phrase. Number of syllables in the current intermediate phrase. Position of the current word in the current phrase. Number of words in the current phrase. Position of the current word in the current intermediate phrase. Number of words in the current intermediate phrase. Position of the current syllable in the utterance. Number of words in the current phrase. Position of the current intermediate phrase in the current phrase. Number of intermediate phrases in the current phrase. Syllable section in the current word (silence, single, begin, middle, end). Syllable section in the current intermediate phrase (silence, single, begin, middle, end). Word section in the current intermediate phrase (silence, single, begin, middle, end). Word section in the current phrase (silence, single, begin, middle, end). Intermediate phrase section in the current phrase (silence, single, begin, middle, end). Phase section in the utterance (silence, single, begin, middle, end). |

3.1.2. Sampling-Based Representation of $F_0$

Several studies have used parametric models, such as the Fujisaki model and DCT [26,39,40]. Li et al. [39] used the Fujisaki model to generate contours of $F_0$ without any modification, but this model does not perform as well as rule-based and frame-based approaches. This suggests that the parameters of the Fujisaki model are complex and challenging to predict. Ronanki et al. [40] used coefficients of DCT to represent the template of an $F_0$ contour, but this approach still requires frame-level features to generate a smooth output of the $F_0$ contour.

Instead of converting values of $F_0$ into parameters of other domains (e.g., the Fujisaki model and DCT), we propose a sampling-based approach to represent the $F_0$ contour. The main idea of the sampling-based model of $F_0$ is to represent the $F_0$ contour within syllables by sampling it with an appropriate number of points. Dynamic features are included to provide temporal information concerning values of $F_0$ in the given syllable and adjacent syllables. The use of dynamic features guarantees that the sampling-based method can produce a smooth output of the $F_0$ contour. The output of the process is used as features for training the RNN. Details of the proposed method of representation of $F_0$ are described as follows.

Before modeling a given $F_0$ contour, interpolation and smoothing must be performed. The interpolation process fills artificial values of $F_0$ between its intermediate values where unvoiced speech segments or short pauses occur. The unusual values of $F_0$ (e.g., error points and micro-prosody) from regions of unvoiced speech are eliminated automatically using the phoneme region described in
the label files. Unvoiced speech segments are interpolated using piecewise cubic interpolation [41] and smoothed by a median filter. Note that we manipulate \( F_0 \) in the logarithmic domain.

Then, values of \( F_0 \) in each syllable are sampled. The \( F_0 \) contour of utterance \( F = [f_1, f_2, \ldots, f_T] \) is represented by concatenating the sequence of the sampled points of \( N \) syllables \( C = [c_1 c_2 \ldots c_N] \). The values of \( F_0 \) in each syllable are sampled as follows:

\[
C_i = \begin{bmatrix} f(b_i + D_i/K) & f(b_i + 2D_i/K) & \cdots & f(b_i + (K-2)D_i/K) & f(b_i + (K-1)D_i/K) & f(b_i + D_i) \end{bmatrix}
\]

where \( f \) denotes the smoothed \( F_0 \) contour, \( b_i \) and \( D_i \) are the starting frame position and duration of syllable \( i \), respectively, and \( K \) is the number of sampling points per syllable. The output vector \( C \) contains the \( KN \) of the sampled values of \( F_0 \).

To improve the performance of the model, additional features are used because using only the sampled values of \( F_0 \) does not guarantee the continuity of the \( F_0 \) contour between adjacent points of syllables. Thus, we use dynamic features computed from the sequence of sampled values of \( F_0 \). We expect dynamic features within and across syllables to improve the continuity of the generated \( F_0 \) contour. The sampled \( F_0 \) vector with dynamic features \( O = [o_1, o_2, \ldots, o_{KN}] \) contains a sequence of sampled values of \( F_0 \), including delta and delta–delta, as follows:

\[
O = \begin{bmatrix} C & \Delta C & \Delta^2 C \end{bmatrix}
\]

\[
= \begin{bmatrix} CW_0 & CW_1 & CW_2 \end{bmatrix}
\]

where \( C \) is the vector of the sampled sequence of log \( F_0 \) and \( W_n \) is a window matrix for calculating the \( n \)-th dynamic feature described in [42]. Syllable-level features \( Y = [y_1, \ldots, y_N] \) are prepared by reshaping the sampled \( F_0 \) vector \( O \) as follows:

\[
y_i = [o_{(i-1)K+1}, o_{(i-1)K+2}, \ldots, o_{iK}] .
\]

In our method of representation, an appropriate number of sampling points \( K \) is required because an inappropriate \( K \) can eliminate detail or create unnecessary values of \( F_0 \). We tuned the model by setting \( K \) to the approximately mean of the duration of all syllables in the Isarn speech corpus. We also explore the effects of the number of sampling points in Section 4.4.3.

3.1.3. RNN Training

We use the RNN to map input features to output features. The RNN has been proposed to overcome the limitations of a feedforward neural network (FFNN) that ignores temporal information in sequential data. In the RNN, information from previous time steps is considered as input to the next time step. Given a sequence of input feature vectors, \([x_1, \ldots, x_T]\), and a sequence of output feature vectors, \([y_1, \ldots, y_T]\), the RNN computes the hidden state vectors and output vectors for a given input sequence as follows:

\[
h_t = \sigma_h(W_{xh}x_t + W_{hh}h_{t-1} + b_h)
\]

\[
y_t = \sigma_y(W_{hy}h_t + b_y)
\]

where, \( x_t, y_t, \) and \( h_t \) are the input vector, output vector, and hidden state vector, respectively, at time \( t \). \( \sigma_h \) is the activation function of the hidden layer, \( \sigma_y \) is the activation function of the output layer, \( W_{xh} \) denotes the weight matrix between the input and the output layers, \( W_{hh} \) is the weight matrix between consecutive hidden states, \( W_{hy} \) is the weight matrix between the hidden and the output layers, and \( b_h \) and \( b_y \) are the bias vectors of the hidden layer and the output layer, respectively.

Typically, the conventional RNN can use only information from the past. The bi-directional RNN has been developed to use information from past and future inputs. It outperforms the unidirectional RNN in many tasks, such as the front-end of text-to-speech systems [43], speech recognition [44],
speech synthesis [11], and machine translation [45,46]. The bi-directional RNN processes the input sequence forward and backward to capture past and future information, respectively, in each layer. Then, the two hidden states are concatenated to produce the output. The iterative process is as follows:

\[ \overrightarrow{h}_t = \sigma_h(\overrightarrow{W} x_t + \overrightarrow{W}_{\overrightarrow{h}} \overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}) \]  
\[ \overleftarrow{h}_t = \sigma_h(\overleftarrow{W} x_t + \overleftarrow{W}_{\overleftarrow{h}} \overleftarrow{h}_{t-1} + b_{\overleftarrow{h}}) \]  
\[ y_t = \sigma_y(\overrightarrow{W}_{\overrightarrow{h} y} \overrightarrow{h}_1 + \overleftarrow{W}_{\overleftarrow{h} y} \overleftarrow{h}_1 + b_y) \]

where \( \overrightarrow{h} \) and \( \overleftarrow{h} \) denote the forward and backward hidden state vector sequences, respectively.

In practice, the performance of the RNN is limited when modeling long-term dependencies in sequential features, called the vanishing gradient problem. We use the recently proposed recurrent units, such as the gated recurrent unit (GRU) [45] and recurrent long short-term memory (LSTM) unit [47], to solve this problem. Based on past work [48], we employed the recurrent bidirectional LSTM (BLSTM) unit.

To train the model, the input features are a sequence of linguistic features. Each input feature vector contains 153-dimensional linguistic features extracting from linguistic features in Table 1. The output feature vector includes 3K-dimensional of sampled \( F_0 \) values with dynamic features, where \( K \) is the number of sampling points per syllable. Theoretically, contextual information is modeled by the internal connection of the recurrent model structure. We thus seek to eliminate contextual information and include only tone for context. The output feature is a sequence of \( F_0 \) vectors and its dynamic future sampled from the original \( F_0 \) contour. Both the input and output features are normalized to have zero mean unit variance.

The hyper-parameters (i.e., the number of hidden layers, number of hidden units, and learning rate) of all models were tuned to achieve close to optimal results on the development set. The model weights were optimized using the Adam-based back-propagation algorithm [49]. To avoid over-fitting, we applied early stopping criteria to stop training when the validation loss had stopped decreasing in 10 consecutive epochs. The maximum number of epochs was set to 150. All models in this work were implemented by using the Keras framework with TensorFlow as back-end [50,51].

3.2. Synthesis of \( F_0 \) Contour

To generate the \( F_0 \) contour, the input text is converted into linguistic features using a text analysis module and these features are used to predict the duration of the phonemes using the RNN-based duration model. Features of duration are included in linguistic features and the input features are fed to the trained RNN-based \( F_0 \) model to obtain the sequence of output feature vectors \( \hat{Y} = [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_N] \). Then, the predicted output features comprising the sampled values of \( F_0 \) and dynamic features are transformed into a feature sequence \( \hat{O} = [\hat{o}_1, \hat{o}_2, \ldots, \hat{o}_{NK}] \), and the \( F_0 \) contour is generated using a parameter generation algorithm [52] with global variance computed on log \( F_0 \). Finally, the smoothed \( F_0 \) vector is scaled to the duration of the syllable. Note that the proposed model generates a continuous \( F_0 \) contour. Voiced/unvoiced flags are obtained from the baseline RNN-based acoustic model described in Section 4.5.  

4. Experiments and Results

In this section, we describe the speech corpus and method of feature extraction and the construction of the proposed model, the conventional frame-based model, and DCT-based models of \( F_0 \), where the last two were employed as the baseline. The performance of the proposed model and baseline models was measured in terms of objective and subjective evaluations.
4.1. Speech Corpus and Feature Extraction

An Isarn speech corpus contains 4700 utterances uttered by one male native Isarn speaker (five hours of speech) [5]. The speech corpus was carefully uttered in the reading style by using text gathered from many sources, such as news articles, web pages, and web boards. The statistical information of utterances in the speech corpus is given in Table 2. The total number of syllables for each tone is shown in Table 3. We divided the speech data into three subsets: 3980 utterances for training, 300 utterances for validation, and 420 utterances for testing. We used a sampling rate of 32,000 Hz instead of the 16,000 Hz used in other studies because this does not degrade the quality of the synthetic speech and is equivalent to using higher sampling rates [53]. These acoustic features were extracted using the WORLD vocoder [54] with a 5 ms frame shift. The acoustic features consisted of three parts: The Mel-cepstral coefficients, band aperiodicity, and log $F_0$.

| Description                           | Values         |
|---------------------------------------|----------------|
|                                      | Training       | Validation | Test        |
| Number of syllables                  | 62,005         | 4619       | 6755        |
| Number of words                      | 51,098         | 3815       | 5580        |
| Number of phrases                    | 7389           | 565        | 826         |
| Number of intermediate phrases       | 3784           | 260        | 374         |
| Length of utterance (phrase)         | 1.86 ± 0.77    | 1.92 ± 0.76| 1.97 ± 0.83 |
| Length of phrase (syllable)          | 12.26 ± 5.59   | 12.41 ± 5.29| 12.61 ± 5.84|
| Length of word (syllable)            | 1.21 ± 0.48    | 1.21 ± 0.49| 1.21 ± 0.48 |
| $F_0$ (Hz)                            | 111.80 ± 19.27 | 111.56 ± 19.22| 111.90 ± 19.39|

Table 2. Statistical information of training, validation, and test sets.

| Tone                   | Number of Syllables (%) |
|------------------------|-------------------------|
|                        | Training                | Validation | Test        |
| Mid-tone               | 14,004 (22.59)          | 1055 (22.84)| 1572 (23.27)|
| Low tone               | 10,448 (16.85)          | 775 (16.78) | 1155 (17.10)|
| Mid-falling tone       | 9522 (15.36)            | 705 (15.26) | 1056 (15.63)|
| High-falling tone      | 11,680 (18.84)          | 906 (19.61) | 1209 (17.90)|
| High tone              | 7955 (12.83)            | 558 (12.08) | 852 (12.61) |
| Rising tone            | 8396 (13.54)            | 620 (13.42) | 911 (13.49) |

Table 3. Total number of syllables for each tone in training, validation, and test sets.

4.2. Optimization and Evaluation Metrics

To evaluate and optimize the performance of the model, the root mean-squared error (RMSE) and correlation (CORR) were used. These metrics were used by considering only the frame in which $F_0$ extracted from natural speech and its predicted value were voiced. Hence, they were modified as:

$$\text{RMSE} = \sqrt{\frac{\sum_{t \in V} (f_t - \hat{f}_t)^2}{|V|}}$$

$$\text{CORR} = \frac{\sum_{t \in V} (f_t - \mu f)(\hat{f}_t - \mu \hat{f})}{\sqrt{\sum_{t \in V} (f_t - \mu f)^2} \sqrt{\sum_{t \in V} (\hat{f}_t - \mu \hat{f})^2}}$$

where $V$ is the set of time indices where both the extracted $F_0$ for natural speech and its predicted value were voiced. $f_t$ and $\hat{f}_t$ denote the extracted $F_0$ and the predicted $F_0$, respectively. $\mu f$ and $\mu \hat{f}$ are mean values of the extracted and predicted $F_0$, respectively. $T$ is the total number of frames. The average
values of RMSE and CORR on the test set were used as objective metrics. A lower RMSE and higher CORR indicated better prediction performance.

4.3. Baseline Systems

In comparison with recent proposals [11,12,38], RNN-based models are significantly better than HMM-based and DNN-based models. Therefore, we used the RNN-based models as a baseline. In terms of the representation of $F_0$, the DCT-based model was used as a baseline because of its advantages reported in [23]. Details of the implementation of the baselines are described below.

4.3.1. Frame-Based Model

The frame-based model was trained [11,12] to generate only $F_0$ because the performance of the $F_0$ model degrades when $F_0$ and the spectral parameters are modeled simultaneously [55]. The input features used consisted of the same set employed on the model for generating speech, whereas the output features consisted of log $F_0$ with dynamic features and a voiced/unvoiced flag. Similar to [11], we included silence in all frames to maintain the continuity of the $F_0$ contour within a sentence. The hyper-parameters were tuned as in the proposed model. The network structure with the best performance consisted of three feedforward layers with 256 nodes per layer, where the top two hidden layers had a BLSTM structure, and each combined 128 forward units with 128 backward units. The model was trained with a learning rate of 0.0001 and had a mini-batch size of 64.

4.3.2. DCT-Based Model

We also trained the model by using DCT coefficients to represent the $F_0$ contour within the syllable. This model and the proposed model were trained using the same input features but their output features were different. Based on past studies [23,24], we used 10 DCT coefficients $C = [c_0, c_1, ..., c_9]$ to represent the $F_0$ contour of each syllable, where $c_0$ represents the mean of $F_0$ over the syllable and the other coefficients represent the curve of $F_0$ within the syllable. The network structure with the best performance consisted of three feedforward layers with 512 nodes per layer, where the top two hidden layers had a BLSTM structure, and each combined 128 forward units with 128 backward units. As with the frame-based model, the model was trained with a learning rate of 0.0001 and a mini-batch size of 64.

4.4. Proposed Model Construction

In this section, we describe the construction of the proposed model and analyze several factors to achieve the best performance. These factors include architectures for the training model, linguistic features to feed as input features, and the number of sampling points per syllable used as output features.

4.4.1. Analysis of Model Architectures

We explored the many-to-many LSTM model in three deferent architectures: Stack bi-directional LSTM (SBLSTM), feedforward followed by bi-directional LSTMs (FF-BLSTM), and bi-directional LSTM followed by feedforward (BLSTM-FF), as shown in Figure 4. The structure of the SBLSTM as presented in prevalent work was used [12,56]. It employed only the BLSTM as the hidden layer. The FF-BLSTM and BLSTM are hybrids of the feedforward and BLSTM layers, respectively. The FF-BLSTM has lower hidden layers with a feedforward structure that is cascaded with the upper hidden layers through a BLSTM adopted from work on text-to-speech systems [11]. The BLSTM-FF uses the lower layer as a BLSTM and the upper layer as a feedforward structure. All models used linear activation in the output layer. We trained the models with the learning rate of 0.0001, a mini-batch size of 64. The best model for each architecture can be summarized as follows:
• SBLSTM: Three hidden layers of BLSTMs, each comprising a forward and a backward layer with 64 units;
• FF-BLSTM: Two feedforward layers with 256 units per layer, where the top layer consisted of three BLSTM layers, each of which in turn consisted of 128 forward and backward units;
• BLSTM-FF: Two BLSTM layers, each consisting of 64 forward and backward units, where the top layer consisted of two feedforward layers with 512 units per layer.

![Diagram of model architecture](image)

**Figure 4.** Model architecture. (a) Stack bi-directional long-short term memory (SBLSTM). (b) FF-BLSTM: Feedforward followed by bidirectional LSTM (BLSTM). (c) LSTM-FF: BLSTM followed by feedforward.

The RMSE and CORR of the best model in each network architecture are shown in Table 4. Note that the voiced/unvoiced flags were obtained from the baseline RNN-based speech synthesizer, the details of implementation of which are described in Section 4.5. As shown in Table 4, the number of model parameters of the models are quite different because we varied the number of hidden layers and units and selected the best performance for each network architecture. Table 4 shows that the SBLSTM architecture delivered the poorest performance, while the FF-BLSTM and BLSTM-FF architectures performed similarly. This indicates that using hybrid feedforward and LSTM layers could improve model performance. We selected the FF-BLSTM for the next evaluation.

**Table 4.** Objective results of best model in each network architecture (± denotes standard deviation and RMSE denotes root mean-squared error).

| Model Architecture | Number of Parameters (Million) | RMSE (Hz)         | CORR               |
|--------------------|--------------------------------|-------------------|--------------------|
| SBLSTM             | 0.32                           | 8.306 ± 2.71      | 0.924 ± 0.04       |
| FF-BLSTM           | 1.32                           | 8.172 ± 2.65      | 0.927 ± 0.04       |
| BLSTM-FF           | 0.60                           | 8.167 ± 2.76      | 0.925 ± 0.04       |

### 4.4.2. Analysis of Linguistic Features

We investigated the influence of each linguistic feature set on predictive performance. We trained the \( F_0 \) models by using several combinations of linguistic feature sets. The best-performing RNN architecture (FF-BLSTM) with the same hyper-parameters was used. The RMSE and CORR of the models trained by different combinations of feature sets are shown in Table 5. The results show that the tonal feature set substantially improved prediction performance (PH_TN was better than PH). However, the combination of all features (PH_TN_PS_DU) achieved the best performance. This indicates that including a combination of phone and tonal feature sets improved prediction performance.
Table 5. The RMSE and correlation (CORR) of models trained by different combinations of feature sets (± denotes standard deviation).

| Model Name | Feature Set                                      | RMSE (Hz)   | CORR       |
|------------|--------------------------------------------------|-------------|------------|
| PH         | only phone features                              | 11.395 ± 3.29 | 0.824 ± 0.12 |
| PH_TN      | phone + tone features                            | 8.825 ± 2.92 | 0.916 ± 0.04 |
| PH_TN_PS   | phone + tone + positional features               | 8.778 ± 2.85 | 0.919 ± 0.04 |
| TN_PS_DU   | tone + positional + duration features            | 8.657 ± 2.69 | 0.918 ± 0.04 |
| PH_TN_PS_DU| all features                                     | 8.172 ± 2.65 | 0.927 ± 0.04 |

4.4.3. Analysis of Number of Sampling Points

We examined the effect of the sampling rate ($K$) of $F_0$ on prediction performance. Based on the distribution of the duration of syllables in Figure 5, we hypothesized that the appropriate value of $K$ would be close to the mean of syllable duration. To test this, we trained the model by varying the value of $K$. The best model of the FF-BLSTM with the same hyper-parameters was used. The RMSE and CORR values are shown in Table 6. The model trained with $K = 45$ gave the best performance in terms of RMSE while the CORRs of all models were similar.

Figure 5. Distribution of syllable duration in Isarn speech corpus.

Table 6. Objective results of the training of the proposed model with different values of $K$.

| $K$ (Frames) | RMSE (Hz)   | CORR       |
|--------------|-------------|------------|
| 10           | 8.320 ± 2.86 | 0.927 ± 0.04 |
| 20           | 8.242 ± 2.76 | 0.927 ± 0.04 |
| 30           | 8.202 ± 2.79 | 0.928 ± 0.04 |
| 40           | 8.201 ± 2.76 | 0.927 ± 0.04 |
| 45           | 8.172 ± 2.65 | 0.927 ± 0.04 |
| 50           | 8.243 ± 2.79 | 0.927 ± 0.04 |
| 60           | 8.315 ± 2.81 | 0.926 ± 0.04 |
| 80           | 8.328 ± 2.65 | 0.927 ± 0.04 |

4.5. Speech Generation

To measure the perceptual performance of the models, the generation of the synthetic speech is required. We employed the FF-BLSTM model to predict spectral features based on [11]. The model was trained by using the input linguistic features adopted from the question set for training the model for HMM-based speech synthesis for Isarn [5]. The input feature vector consisted of 489-dimensional linguistic features: 472 dimensions of categorical linguistic contexts (e.g., phonemes identities, tone of syllable), 14 dimensions of numerical linguistic contexts (e.g., position of the current syllable in the current word, number of syllables in the current word), and 3 dimensions of frame-level features.
The frame-level input features were considered, including forward/backward positions of the given frame in the given phone and phone duration. The output feature vector comprised 196-dimensional acoustic features containing the 60-dimensional Mel-cepstral coefficients, 4-dimensional band aperiodicities, and log $F_0$ with their dynamic features and voiced/unvoiced flags. Similar to [11], we included all silence frames for training to preserve the continuity of acoustic features within a sentence. The input and output features were normalized to have zero mean and unit variance. We used the network structure of three feedforward layers, with 512 nodes per layer, where the top two hidden layers had a BLSTM structure. Each combined 128 forward and 128 backward nodes. This model was used to generate the speech parameters for all experiments.

In the synthesis stage, the sequence of input feature vectors was fed to the trained RNN-based acoustic model to produce the speech parameters. Then, these speech parameters were smoothened by using a parameter generation algorithm [52]. Following this, the output speech parameters were enhanced using a post-filtering algorithm [57] to improve the naturalness of the synthetic speech. Finally, the waveform of speech was generated through the speech vocoder.

4.6. Objective Evaluation

The objective evaluation measured the distortion between the original and generated $F_0$ contours in terms of RMSE and CORR. The values for the proposed model named sampling (SAMP)-based model and baselines are shown in Table 7. The generation time of each model used for generating $F_0$ contour for all utterances in the test set is also reported. The DCT-based model gave the poorest performance in terms of both RMSE and CORR. The SAMP-based model outperformed the frame-based model in terms of RMSE while their CORR values were similar. It can be found that the generation time of the SAMP-based model is lower than the Frame-based model although the SAMP-based model have a larger number of parameters than the Frame-based model. The possible reason is the Frame-based model generates $F_0$ values frame by frame while the SAMP-based model generates $F_0$ syllable by syllable. The generation time of the DCT-based model is comparable to the Frame-based model because it requires the additional time for converting DCT-coefficients to $F_0$ contour.

However, we noticed that the RMSE and CORR of the models were slightly different. In this case, the perceptual evaluation is required to further measure the performance of the models because the objective result might not always be well correlated with perception of the listener [58].

| Model                  | Number of Parameters (Million) | Generation Time (s) | RMSE (Hz)        | CORR          |
|------------------------|-------------------------------|---------------------|-----------------|---------------|
| Frame based            | 1.04                          | 35                  | $8.362 \pm 3.35$ | $0.928 \pm 0.04$ |
| Discrete cosine transform (DCT) based | 1.65                        | 36                  | $8.762 \pm 2.71$ | $0.915 \pm 0.04$ |
| SAMP based             | 1.32                          | 10                  | $8.172 \pm 2.65$ | $0.927 \pm 0.04$ |

4.7. Subjective Evaluation

Typically, the objective evaluation is useful for training the model but does not reflect the perception of the listener [58]. Thus, we also conducted tests of subjective preference. As these tests were used to investigate the generation of $F_0$ by the models, the spectral parameters were generated using the same model, whereas $F_0$ was generated using different models. To force the listener to concentrate on the generation of $F_0$, the phone duration was obtained from the transcription files because prosody is also dependent on the performance of the duration model.

A total of 30 native speakers participated in each test. All listeners speak fluently in the central Isarn dialect. In each test, the subjects were asked to listen to 20 pairs of utterances (some samples of synthetic speech with three $F_0$ models are available at https://isarn-samp-f0.github.io) randomly selected from the test set, and determined the item in each pair that sounded more natural, or chose a “no preference” option if they found the two utterances to be very similar. The order of the speech samples in each pair was swapped. The listeners were allowed to play back the recordings of the
utterances as many times as they wanted before assigning a score to them. A t-test was used to show that the differences between the compared systems were significant ($p < 0.01$).

Three preference tests were conducted consisting of comparisons between the frame-based model and DCT-based model, between the SAMP-based model and frame-based model, and between the DCT-based model and SAMP-based model. Figure 6 shows the preference scores of the system pairs. It is clear that the preference score of the frame-based model was lower than those of the DCT-based and SAMP-based models, which were trained using syllable-level features, although the DCT-based model had recorded a poorer performance in the objective evaluation. This indicates that using syllable-level features is effective for learning the complex variations in tonal contours. However, the difference between the frame-based and DCT-based models was not significant ($p = 0.3067$). Considering the representation of $F_0$, the SAMP-based model was significantly better than the DCT-based model ($p = 0.005$). This indicates that the sampling-based method can provide a better representation of the $F_0$ contour for the Isarn speech corpus.

![Figure 6](image1)

*Figure 6.* Preference scores with 99% confidence for each system pair: (a) Frame-based model versus DCT-based model, (b) SAMP-based model versus frame-based model, and (c) SAMP-based model versus DCT-based model.

To demonstrate the effect of using the proposed syllable-level features, Figure 7 shows a comparison of the reference $F_0$ contour and the $F_0$ contours generated by the three systems using the sentences “เรื่องมีอยู่ในจินตนาการของคุณ” / ‘-Rayjong mii ooy in jotnaagara of koon” (“Hey, come to see, I am checking for counterfeit money.” in English translation) and “当你的水牛给出生，你不要忘记要它的胎盘。” / daw ngk’uaj woon gai soong laak l'ap jie hoom fa'aw nkg’uaj nam d'aa/ (“When your buffalo gives birth, you don’t forget to get its placenta.” in English translation). As shown in Figure 7a, the generated $F_0$ contour using the proposed model was more appropriate than those of the baseline systems at both the syllable level (e.g., from frame 300 to 350) and the utterance level (e.g., from 75
to 275). Figure 7b demonstrates inappropriate contours of $F_0$ generated by the DCT-based method (e.g., frame 250 to 300). These results were obtained because the DCT-based method might have generated sub-par values for some coefficients that caused the overall $F_0$ contour to deviate.

Figure 7. Comparison of $F_0$ contours generated using frame-based, DCT-based, and SAMP-based models, (a) the sentence means “Hey, come to see, I am checking for counterfeit money.” in English, (b) the sentence means “When your buffalo gives birth. You don’t forget to get its placenta.” in English.

5. Conclusions and Future Work

In this paper, we proposed an RNN-based $F_0$ model for Isarn speech synthesis. The model can generate $F_0$ contours at the syllable level instead of the frame level. The $F_0$ contour within the syllable was represented by sampled values of $F_0$ and their dynamic features. To achieve the best performance, we investigated the performance of several model architectures: The SBLSTM, FF-BLSTM, and BLSTM-FF. Based on an objective test, a hybrid of feedforward and BLSTM delivered the best performance. We compared the optimized model with the frame-based model and the DCT-based model in terms of the representation of $F_0$. The objective results of the proposed method and baseline were slightly different. However, the results of subjective tests showed that the proposed model significantly outperformed the baseline systems. This suggests that modeling $F_0$ at the syllable level using the proposed sampling-based method of representation of $F_0$ was effective for learning the complex variation in tonal contours. In future work, we will emphasize the generation of phoneme duration to improve the naturalness of synthesized speech.

Author Contributions: Conceptualization, Investigation, PJ, and PS; Methodology, PJ, and PS; Project administration, PS; Resources, PJ, and PS; Software, PJ, and PS; Supervision, PS; Writing—original draft, PJ; Writing—review and editing, PJ, and PS. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by a Computer and Information Science Interdisciplinary research grant from the Department of Computer Science, Faculty of Science, Khon Kaen University, Khon Kaen, Thailand.

Acknowledgments: This work was supported through a Computer and Information Science Interdisciplinary research grant from the Department of Computer Science, Faculty of Science, Khon Kaen University, Khon Kaen, Thailand.

Conflicts of Interest: The authors declare no conflict of interest.
References

1. Seresangtakul, P.; Takara, T. Synthesis of Polysyllabic Sequences of Thai Tones Using a Generative Model of Fundamental Frequency Contours. *IEICE Trans. Inf. Syst.* 2005, 125, 1101–1108. [CrossRef]

2. Chomphan, S.; Kobayashi, T. Implementation and evaluation of an HMM-based Thai speech synthesis system. In Proceedings of the Eighteen Annual Conference of the International Speech Communication Association, Antwerp, Belgium, 27–31 August 2007; pp. 2849–2852.

3. Qian, Y.; Soong, F.; Chen, Y.; Chu, M. An HMM-Based Mandarin Chinese Text-To-Speech System. In Proceedings of the 5th International Symposium on Chinese Spoken Language Processing, Singapore, 13–16 December 2006; pp. 223–232.

4. Vu, T.; Luong, C.; Nakamura, S. An HMM-based Vietnamese speech synthesis system. In Proceedings of the Oriental COCOSDA International Conference on Speech Database and Assessments, Urumqi, China, 10–12 August 2009; pp. 116–121.

5. Janyoi, P.; Seresangtakul, P. Isarn Dialect Speech Synthesis using HMM with syllable-context features. *ECTI Trans. Comput. Inf. Technol.* 2018, 12, 81–89.

6. Wang, C.C.; Ling, Z.H.; Zhang, B.F.; Dai, L.R. Multi-Layer F0 Modeling for HMM-Based Speech Synthesis. In Proceedings of the 6th International Symposium on Chinese Spoken Language Processing, Kunming, China, 16–19 December 2008; pp. 1–4.

7. Lei, M.; Wu, Y.J.; Soong, F.K.; Ling, Z.H.; Dai, L.R. A hierarchical F0 modeling method for HMM-based speech synthesis. In Proceedings of the 11th Annual Conference of the International Speech Communication Association, Chiba, Japan, 26–30 September 2010; pp. 2170–2173.

8. Chomphan, S.; Kobayashi, T. Tone correctness improvement in speaker dependent HMM-based Thai speech synthesis. *Speech Commun.* 2008, 50, 392–404. [CrossRef]

9. Wu, Y.J.; Soong, F. Modeling pitch trajectory by hierarchical HMM with minimum generation error training. In Proceedings of the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 25–30 March 2012; pp. 4017–4020.

10. Zen, H.; Senior, A.; Schuster, M. Statistical parametric speech synthesis using deep neural networks. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 7962–7966.

11. Fan, Y.; Qian, Y.; Xie, F.L.; Soong, F.K. TTS synthesis with bidirectional LSTM based recurrent neural networks. In Proceedings of the 15th Annual Conference of the International Speech Communication Association, Singapore, 14–18 September 2014; pp. 1964–1968.

12. Fernandez, R.; Rendel, A.; Ramabhadran, B.; Hoory, R. Prosody contour prediction with long short-term memory, bi-directional, deep recurrent neural networks. In Proceedings of the 15th Annual Conference of the International Speech Communication Association, Singapore, 14–18 September 2014; pp. 2268–2272.

13. Zen, H.; Sak, H. Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Queensland, Australia, 19–24 April 2015; pp. 4470–4474.

14. Wang, Y.; Skerry-Ryan, R.J.; Stanton, D.; Wu, Y.; Weiss, R.J.; Jaitly, N.; Yang, Z.; Xiao, Y.; Chen, Z.; Bengio, S.; et al. Tacotron: Towards End-to-End Speech Synthesis. In Proceedings of the 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, 20–24 August 2017; pp. 4006–4010.

15. Ping, W.; Peng, K.; Gibiansky, A.; Arik, S.O.; Kannan, A.; Narang, S.; Raiman, J.; Miller, J.L. Deep Voice 3: Scaling Text-to-Speech with Convolutional Sequence Learning. In Proceedings of the 6th International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018; pp. 1–16.

16. Chung, Y.A.; Wang, Y.; Hsu, W.N.; Zhang, Y.L.; Skerry-Ryan, R.J. Semi-supervised Training for Improving Data Efficiency in End-to-end Speech Synthesis. In Proceedings of the 44th IEEE International Conference on Acoustics, Speech and Signal Processing, Brighton, UK, 12–17 May 2019; pp. 6940–6944.

17. Thubthong, N.; Kijsirikul, B.; Luksaneeyanawin, S. Tone recognition in Thai continuous speech based on coarticulation, intonation and stress effects. In Proceedings of the 7th International Conference on Spoken Language Processing, Denver, CO, USA, 16–20 September 2002; pp. 1169–1172.

18. Gandour, J.; Potsisuk, S.; Dechongkit, S. Tonal Coarticulation in Thai. *J. Phon.* 1994, 22, 477–492. [CrossRef]
19. Taylor, P. Analysis and synthesis of intonation using the Tilt model. *J. Acoust. Soc. Am.* 2000, 107, 1697–1714. [CrossRef]

20. Thangthai, A.; Thatphithakkul, N.; Wutiwiwatchai, C.; Rugchatjaroen, A.; Saychum, S. T-Tilt: A modified Tilt model for F0 analysis and synthesis in tonal languages. In Proceedings of the 9th Annual Conference of the International Speech Communication Association, Brisbane, Australia, 22–26 September 2008; pp. 2270–2273.

21. Fujisaki, H.; Hirose, K. Analysis of voice fundamental frequency contours for declarative sentences of Japanese. *J. Acoust. Soc. Jpn. (E)* 1984, 5, 233–242. [CrossRef]

22. Xu, Y.; Wang, E. Pitch targets and their realization: Evidence from Mandarin Chinese. *Speech Commun.* 2001, 33, 319–337. [CrossRef]

23. Teutenberg, J.; Watson, C.; Riddle, P. Modelling and synthesising F0 contours with the discrete cosine transform. In Proceedings of the 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, Las Vegas, NV, USA, 30 March–4 April 2008; pp. 3973–3976.

24. Stan, A.; Giurgiu, M. A superpositional model applied to F0 parameterization using DCT for text-to-speech synthesis. In Proceedings of the 6th Conference on Speech Technology and Human–Computer Dialogue, Brasov, Romania, 18–21 May 2011; pp. 1–6.

25. Luo, L.; Xian, X. Integration of Intonation in Trainable Speech Synthesis. In Proceedings of the 4th International Conference on Speech Prosody, Campinas, Brazil, 6–9 May 2008; pp. 75–78.

26. Hirose, K.; Eto, M.; Minematsu, N.; Sakurai, A. Corpus-based synthesis of fundamental frequency contours based on a generation process model. In Proceedings of the Seventh European Conference on Speech Communication and Technology, Aalborg, Denmark, 3–7 September 2001; pp. 2255–2258.

27. Wu, Z.; Qian, Y.; Soong, F.K.; Zhang, B. Modeling and Generating Tone Contour with Phrase Intonation for Mandarin Chinese Speech. In Proceedings of the 2008 6th International Symposium on Chinese Spoken Language Processing, Tainan, Taiwan, 29 November–3 December 2010; pp. 144–147.

28. Li, Y.; Tao, J.; Hirose, K.; Xu, X.; Lai, W. Hierarchical stress modeling and generation in mandarin for expressive Text-to-Speech. *Speech Commun.* 2015, 72, 59–73. [CrossRef]
41. Fujisaki, H.; Narusawa, S.; Maruno, M. Pre-processing of fundamental frequency contours of speech for automatic parameter extraction. In Proceedings of the 2000 5th International Conference on Signal Processing Proceedings, Beijing, China, 21–25 August 2000; pp. 722–725.

42. Zen, H.; Tokuda, K.; Kitamura, T. Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic feature vector sequences. *Comput. Speech Lang.* 2007, 21, 153–173. [CrossRef]

43. Le, N.T.; Sadat, F.; Menard, L.; Dinh, D. Low-Resource Machine Transliteration Using Recurrent Neural Networks. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 2019, 18, 13:1–13:14. [CrossRef]

44. Graves, A.; Mohamed, A.; Hinton, G. Speech recognition with deep recurrent neural networks. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, CO, Canada, 26–31 May 2013; pp. 6645–6649.

45. Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 1724–1734.

46. Wang, R.; Zhao, H.; Ploux, S.; Lu, B.L.; Utiyama, M.; Sumita, E. Graph-Based Bilingual Word Embedding for Statistical Machine Translation. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 2018, 17, 31:1–31:23. [CrossRef]

47. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* 1997, 9, 1735–1780. [CrossRef]

48. Al-Radhi, M.S.; Csapo, T.G.; Nemeth, G. Deep Recurrent Neural Networks in Speech Synthesis Using a Continuous Vocoder. In Proceedings of the 19th International Conference on Speech and Computer, Hatfield, Hertfordshire, UK, 12–16 November 2017; pp. 282–291.

49. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.

50. Chollet, F. Keras. Available online: https://github.com/fchollet/keras (accessed on 25 April 2020).

51. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. TensorFlow: A System for Large-scale Machine Learning. In Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, Savannah, GA, USA, 2–4 November 2016; pp. 265–283.

52. Tokuda, K.; Yoshimura, T.; Masuko, T.; Kobayashi, T.; Kitamura, T. Speech parameter generation algorithms for HMM-based speech synthesis. In Proceedings of the 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing, Istanbul, Turkey, 5–9 June 2000; pp. 1315–1318.

53. Stan, A.; Yamagishi, J.; King, S.; Aylett, M. The Romanian speech synthesis (RSS) corpus: Building a high quality HMM-based speech synthesis system using a high sampling rate. *Speech Commun.* 2011, 53, 442–450. [CrossRef]

54. Morise, M.; Yokomori, F.; Ozawa, K. WORLD: A Vocoder-Based High-Quality Speech Synthesis System for Real-Time Applications. *IEICE Trans. Inf. Syst.* 2016, E99.D, 1877–1884. [CrossRef]

55. Chen, B.; Chen, Z.; Xu, J.; Yu, K. An investigation of context clustering for statistical speech synthesis with deep neural network. In Proceedings of the 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, 6–10 September 2015; pp. 2212–2216.

56. Sun, L.; Kang, S.; Li, K.; Meng, H. Voice conversion using deep Bidirectional Long Short-Term Memory based Recurrent Neural Networks. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Queensland, Australia, 19–24 April 2015; pp. 4869–4873.

57. Yoshimura, T.; Tokuda, K.; Masuko, T.; Kobayashi, T.; Kitamura, T. Incorporating a mixed excitation model and postfilter into HMM-based text-to-speech synthesis. *Syst. Comput. Jpn.* 2005, 36, 43–50. [CrossRef]

58. Wu, Z.; King, S. Improving Trajectory Modelling for DNN-Based Speech Synthesis by Using Stacked Bottleneck Features and Minimum Generation Error Training. *IEEE/ACM Trans. Audio Speech Lang. Process.* 2016, 24, 1255–1265. [CrossRef]