Autoregressive neural F0 model for statistical parametric speech synthesis

Autoregressive neural F0 model for statistical parametric speech synthesis

Xin Wang, *Student Member, IEEE,* Shinji Takaki, *Member, IEEE,* Junichi Yamagishi, *Senior Member, IEEE*

***Abstract*—Recurrent neural networks (RNNs) have been successfully used as fundamental frequency (F0) models for text-to-speech synthesis. However, this study showed that a normal RNN may not take into account the statistical dependency of the F0 data across frames and consequently only generate noisy F0 contours when F0 values are sampled from the model. A better model may take into account the causal dependency of the current F0 datum on the previous frames' F0 data. One such model is the shallow autoregressive (AR) recurrent mixture density network (SAR) that we recently proposed. However, as this study showed, an SAR is equivalent to the combination of trainable linear filters and a conventional RNN. It is still weak for F0 modeling.**

To better model the temporal dependency in F0 contours, we propose a deep AR model (DAR). On the basis of an RNN, this DAR propagates the previous frame's F0 value through the RNN, which allows non-linear AR dependency to be achieved. We also propose F0 quantization and data dropout strategies for the DAR. Experiments on a Japanese corpus demonstrated that this DAR can generate appropriate F0 contours by using the random-sampling-based generation method, which is impossible for the baseline RNN and SAR. When a conventional mean-based generation method was used in the proposed DAR and other experimental models, the DAR generated accurate and less over-smoothed F0 contours and achieved a better mean-opinion-score in a subjective evaluation test.

***Index Terms*—fundamental frequency, F0, pitch, speech synthesis, neural network, autoregressive model**

I. Introduction

Text-to-speech synthesis (TTS) converts a text string into a speech waveform. A common TTS system consists of a front-end and back-end [1], where the front-end determines "how to read" and the back-end creates the waveform accordingly. The back-end can be implemented using the statistical parametric speech synthesis (SPSS) framework [2], [3]. Given the linguistic features extracted from the text by the front-end, the SPSS-based back-end uses statistical models to generate compact acoustic features such as the fundamental frequency (F0) and Mel-cepstral features. It then uses a vocoder to convert the acoustic features into a speech waveform.

The F0 is an essential acoustic feature of the speech waveform. It is perceived as pitch and conveys the tone and intonation in an utterance [4], [5]. The SPSS framework uses hidden Markov models (HMMs), decision trees, and various types of neural networks (NNs) [6], [7], [8] to jointly model the F0 and other acoustic features frame-by-frame [2], [3], [9]. Besides the F0 model integrated into the SPSS framework, various stand-alone F0 models have also been proposed on the basis of expert knowledge (e.g., [10], [11]) and statistical approaches (e.g., [12], [13]).

In this study, we focused on NN-based F0 models that can be plugged into the SPSS framework. Although various NN-based models have been investigated [14], [15], [16], [17], [18], [19], they may be imperfect for F0 modeling. In most of these models, the F0 data of different frames are implicitly treated as statistically independent; moreover, this assumption remains even if a recurrent NN (RNN) with long-short memory units (LSTMs) is used. In this paper, we explain the theoretical model assumption of conventional RNNs and the empirical demonstration of it using a technique called random sampling.

A potentially better approach is augmenting a normal RNN with autoregressive (AR) dependency. The idea is to define and learn the F0 distribution conditioned on previous F0 observations in an F0 sequence. One example of such a model is the AR recurrent mixture density network (RMDN) proposed in our previous study [20]. This model summarizes the F0 data of previous frames in the F0 sequence and adjusts the mean of the current F0 distribution using a linear function. However, our current study showed that this AR-based RMDN is equivalent to a combination of a normal RNN and trainable linear filters and only the filters capture the AR dependency. Hence, this model, referred to as the shallow AR model (SAR) in this paper, is still weak for F0 modeling, even though it can reduce the over-smoothing effect on the generated F0 contours.

For this paper, we propose a deep AR model (DAR) toward a better F0 model[1]. Based on a normal RNN, the proposed DAR feeds back a previous F0 observation as the input to a recurrent layer. This feedback link makes it feasible to achieve the non-linear AR dependency over a longer time span. With additional improvements, such as quantized F0 representation and data dropout strategies, the DAR performed significantly better than the RNN and SAR in experiments. Specifically, it increased the dynamic range of the generated F0 contours without decreasing accuracy; thus, improving the perceived naturalness of synthetic speech. Furthermore, the DAR generated good F0 contours via random sampling, which

Xin Wang and Shinji Takaki are with National Institute of Informatics, Japan. e-mail:wangxin@nii.ac.jp, takaki@nii.ac.jp

Junichi Yamagishi is with the National Institute of Informatics, Japan and with the Centre of Speech Technology Research, University of Edinburgh, U.K. e-mail: jyamagis@nii.ac.jp.

The guest editor coordinating the review of this manuscript was Dr. Heiga Zen. This work was partially supported by MEXT KAKENHI Grant Numbers (15H01686, 16H06302, 17H04687).

[1]A part of this study was published in [21]. Compared with the previously published study, we theoretically analyzed the DAR, the SAR, and a baseline RNN in this study and compared them on a larger Japanese corpus through a large-scale listening test. We also obtained new results using a random-sampling-based F0 generation method.

had not been achieved with previous statistical F0 models to the best of our knowledge.

In the rest of this paper, Section II gives an overview of related F0 models. Then, Section III defines and analyzes the SAR, and Section IV defines the proposed DAR and related techniques. Section V shows the details of experiments. Finally, Section VI concludes this paper.

II. Towards AR Neural F0 Model

A. Classical methods for F0 modeling

In this study, we considered F0 modeling at the frame level and assumed that the input linguistic features are given by a TTS front-end. The task of an F0 model is to learn the mapping from the linguistic features to the F0 for each speech frame.

A classical F0-modeling approach may require two steps. It first transforms the F0 data within a segment into a compact parametric form, after which it learns the mapping from the linguistic features to the compact F0 parameters. For example, the parameters of the Fujisaki model [22], [23] can be used to represent the F0 contour within a syllable. Statistical models, such as the decision tree, can then be used to learn the mapping for the second step [24]. This approach can also be applied on the basis of the Tilt model [25], [26], Parallel Encoding and Target Approximation (PENTA) model [27], [28], [29], and other parametric or non-parametric representations, namely the target F0 points [12], [14], coefficients from discrete-cosine transformation [30], [31], [32], [33], [34], [35], and parameters extracted from functional data analysis [36].

Rather than the above-mentioned two-step strategy, it is also possible to directly map the linguistic features to the F0 frame-by-frame. Examples are the multi-space probability distribution HMM [37], continuous F0 model [9], and hierarchical F0 model [38], [39] designed for HMM-based SPSS. Recently, the deep feedforward NN [7] has been used to jointly model the F0 and other spectral features at the frame level. However, it was found that a feedforward NN may prioritize the high-dimensional spectral features over the F0 [40]. An alternative approach is to only model the F0 contours. In this case, the RNN-based F0 model [19] performs quite well.

B. Baseline neural networks for F0 modeling

We focused on NN-based F0 models, the task of which is to convert linguistic features of T frames $\boldsymbol{x}_{1:T} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T\}$ into an F0 sequence $\widehat{\boldsymbol{o}}_{1:T} = \{\widehat{\boldsymbol{o}}_1, \cdots, \widehat{\boldsymbol{o}}_T\}$. The goal is to make $\widehat{\boldsymbol{o}}_{1:T}$ approximate the true F0 sequence $\boldsymbol{o}_{1:T}$. Although F0 is a one-dimensional value, for general explanation, we define that $\widehat{\boldsymbol{o}}_t$ and $\boldsymbol{o}_t$ are D-dimensional real-valued vectors, i.e., $\widehat{\boldsymbol{o}}_t, \boldsymbol{o}_t \in \mathbb{R}^D, D \in \mathbb{N}$.

Let us consider an RNN with a bi-directional recurrent hidden layer and linear output layer. Given the input $\boldsymbol{x}_{1:T}$, this RNN calculates the output in three steps:

$$\boldsymbol{h}_t^{(f)} = \sigma(\boldsymbol{W}_h^{(f)}\boldsymbol{h}_{t-1}^{(f)} + \boldsymbol{W}_i^{(f)}\boldsymbol{x}_t + \boldsymbol{b}_h^{(f)}), \quad (1)$$

$$\boldsymbol{h}_t^{(b)} = \sigma(\boldsymbol{W}_h^{(b)}\boldsymbol{h}_{t+1}^{(b)} + \boldsymbol{W}_i^{(b)}\boldsymbol{x}_t + \boldsymbol{b}_h^{(b)}), \quad (2)$$

$$\boldsymbol{\mathcal{H}}_{\boldsymbol{\Theta}}(\boldsymbol{x}_{1:T}, t) = \boldsymbol{W}_o^{(f)}\boldsymbol{h}_t^{(f)} + \boldsymbol{W}_o^{(b)}\boldsymbol{h}_t^{(b)} + \boldsymbol{b}_o. \quad (3)$$

Here, $\sigma(\cdot)$ is an activation function, $\boldsymbol{h}_t^{(f)}$ and $\boldsymbol{h}_t^{(b)}$ are hidden features computed by the recurrent layer in the forward and backward directions, $\boldsymbol{\Theta} = \{\boldsymbol{W}_h^{(*)}, \boldsymbol{W}_i^{(*)}, \boldsymbol{W}_o^{(*)}, \boldsymbol{b}_h^{(*)}, \boldsymbol{b}_o\}$ denotes network weights, and $\boldsymbol{\mathcal{H}}_{\boldsymbol{\Theta}}(\boldsymbol{x}_{1:T}, t)$ denotes the RNN's output at the t-th frame. The network weight $\boldsymbol{\Theta}$ can be trained by minimizing a mean square error (MSE) $E = \sum_{t=1}^{T} ||\boldsymbol{\mathcal{H}}_{\boldsymbol{\Theta}}(\boldsymbol{x}_{1:T}, t) - \boldsymbol{o}_t||^2$. Given the trained $\boldsymbol{\Theta}^*$, the RNN can generate $\widehat{\boldsymbol{o}}_{1:\tilde{T}}$ for a new input $\tilde{\boldsymbol{x}}_{1:\tilde{T}}$ by setting $\widehat{\boldsymbol{o}}_t = \boldsymbol{\mathcal{H}}_{\boldsymbol{\Theta}^*}(\tilde{\boldsymbol{x}}_{1:\tilde{T}}, t)$, $\forall t \in \{1, \cdots, \tilde{T}\}$.

The above approach of using an RNN as a regression tool is reasonable. It has been shown that the MSE-based training method is equivalent to a maximum-likelihood estimation scheme, i.e., $\boldsymbol{\Theta}^* = \arg\max_{\boldsymbol{\Theta}} \log p(\boldsymbol{o}_{1:T}|\boldsymbol{x}_{1:T}; \boldsymbol{\Theta})$, under the assumption that the probabilistic density function (PDF) $p(\boldsymbol{o}_{1:T}|\boldsymbol{x}_{1:T}; \boldsymbol{\Theta})$ is a product of Gaussian distributions [41]:

$$p(\boldsymbol{o}_{1:T}|\boldsymbol{x}_{1:T}; \boldsymbol{\Theta}) = \prod_{t=1}^{T} p(\boldsymbol{o}_t|\boldsymbol{x}_{1:T}; \boldsymbol{\Theta}) \quad (4)$$

$$= \prod_{t=1}^{T} \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mathcal{H}}_{\boldsymbol{\Theta}}(\boldsymbol{x}_{1:T}, t), \beta\boldsymbol{I}). \quad (5)$$

Here, $\mathcal{N}(\cdot)$ is the Gaussian distribution, $\boldsymbol{I}$ is the identity matrix, and β is a variance parameter that does not affect $\boldsymbol{\Theta}^*$. After $\boldsymbol{\Theta}^*$ is estimated, it is also reasonable to use $\widehat{\boldsymbol{o}}_t = \boldsymbol{\mathcal{H}}_{\boldsymbol{\Theta}^*}(\tilde{\boldsymbol{x}}_{1:\tilde{T}}, t)$ as the generation method because $\boldsymbol{\mathcal{H}}_{\boldsymbol{\Theta}^*}(\boldsymbol{x}_{1:T}, t)$ approximates the true conditional mean $\mathbb{E}[\boldsymbol{o}_t|\tilde{\boldsymbol{x}}_{1:\tilde{T}}]$ [41]. For this reason, this generation method is referred to as the *mean-based generation* method.

Because an RNN's output only approximates $\mathbb{E}[\boldsymbol{o}_t|\tilde{\boldsymbol{x}}_{1:\tilde{T}}]$, it cannot generalize well when the true distribution is multi-modal. A better model is the RMDN [42], [43]. It directly defines the PDF as

$$p(\boldsymbol{o}_{1:T}|\boldsymbol{x}_{1:T}; \boldsymbol{\Theta}) = \prod_{t=1}^{T} \sum_{m=1}^{M} \omega_t^m \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_t^m, \boldsymbol{\Sigma}_t^m), \quad (6)$$

where M is the number of mixture components of a Gaussian mixture model (GMM), and $\boldsymbol{\mathcal{M}}_t = \{\omega_t^1, \cdots, \omega_t^M, \boldsymbol{\mu}_t^1, \cdots, \boldsymbol{\mu}_t^M, \boldsymbol{\Sigma}_t^1, \cdots, \boldsymbol{\Sigma}_t^M\}$ is the GMM's parameter set. The covariance matrices $\{\boldsymbol{\Sigma}_t^1, \cdots, \boldsymbol{\Sigma}_t^M\}$ are usually set to be diagonal. In an RMDN, the value of $\boldsymbol{\mathcal{M}}_t$ is computed by the internal RNN, i.e., $\boldsymbol{\mathcal{M}}_t = \boldsymbol{\mathcal{H}}_{\boldsymbol{\Theta}}(\boldsymbol{x}_{1:T}, t)$. One example of an RMDN is plotted on the left side of Figure 1. As the figure shows, the internal RNN outputs the parameters of the GMMs that describe the distribution of $\boldsymbol{o}_t$. The network weights $\boldsymbol{\Theta}$ can be trained by maximizing the likelihood. During generation, the sequence of $\boldsymbol{\mathcal{M}}_t$ is computed given the input features, and the mean of the mixture component with the largest weight can used as the model's output $\widehat{\boldsymbol{o}}_t$ for each frame [42].

Next, let us consider modeling F0 as quantized rather than continuous values. The theoretic motivation is that humans have difficulty telling two sounds apart with a small difference in frequency, which is known as the just-noticeable difference of pitch [44]. Accordingly, we may quantize F0 values into several bins and treat the quantized bins as discrete values. Then, similar to the idea of modeling quantized speech waveform [45], we can model the quantized F0 using the framework for modeling categorical data.

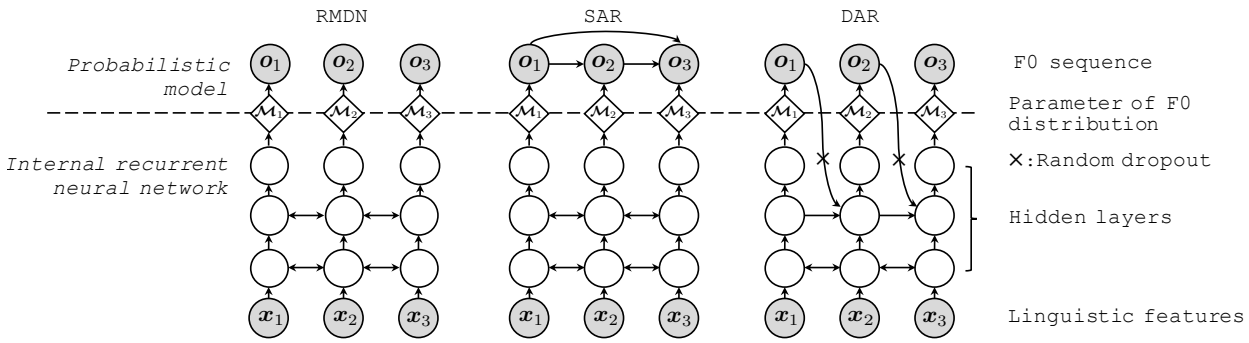


Fig. 1. Illustration of RMDN, SAR and DAR. Grey circles denote observed random variables, while white circles denote deterministic values calculated with the internal RNN. Diamonds, which are the output of the internal RNN, denote the parameter set of distribution for $\boldsymbol{o}_t$. Detailed network structures for the experiments are explained in Section V.

To do this, we can use an RMDN after replacing the GMM with a probabilistic mass function (PMF) based on a multinomial distribution. Suppose $\boldsymbol{o}_t$ is a one-hot vector for a discrete datum, i.e., one bin of the quantized F0, the PMF of $\boldsymbol{o}_{1:T}$ is defined as

$$P(\boldsymbol{o}_{1:T}|\boldsymbol{x}_{1:T};\boldsymbol{\Theta}) = \prod_{t=1}^{T} P(\boldsymbol{o}_t|\boldsymbol{x}_{1:T};\boldsymbol{\Theta}). \tag{7}$$

Here, $P(\boldsymbol{o}_t|\boldsymbol{x}_{1:T};\boldsymbol{\Theta})$ is computed using the internal RNN with a softmax output layer, and $\boldsymbol{\Theta}$ can be estimated by maximizing the likelihood or equivalently minimizing the cross-entropy. This RMDN with a softmax output layer has been used for part-of-speech tagging [46] and name-entity recognition [47], and it was used in this study as a baseline quantized F0 model. During generation, the output F0 can be randomly sampled from the PMF calculated by the softmax layer, or the bin with the largest probability can be directly used as the generated F0. However, because the F0 value is originally continuous, we can also generate the F0 by using the expected values of the quantized F0, which is explained in Section IV-D.

C. General idea of AR neural model

From Equations (5), (6), and (7), we can see that regardless of whether F0 is quantized, it is assumed with the baseline models that $\boldsymbol{o}_{t_1}$ is statistically independent from $\boldsymbol{o}_{t_2}$ given the condition $\boldsymbol{x}_{1:T}$, $\forall t_2 \neq t_1$. This assumption may be inappropriate for F0 modeling because adjacent voiced frames usually have a similar F0 value. Consequently, a baseline F0 model may only learn the linguistic-feature-conditioned F0 'uni-gram', which may be multi-mode or large in variance. If the mean or expected values are used for generation, the generated F0 will approximate $\mathbb{E}[\boldsymbol{o}_t|\boldsymbol{x}_{1:T}]$, but the output F0 contour may be over-smoothed. While an RMDN with a large number of GMM mixtures could better model the F0 'uni-gram' per frame, it may not model the temporal dependency of the F0 values across the frames, as discussed in Section IV-A.

To model the temporal dependency of $\boldsymbol{o}_{1:T}$, a general idea is to re-define the model as a Markov random field where all random variables form a single clique, i.e., $p(\boldsymbol{o}_{1:T}|\boldsymbol{x}_{1:T}) = \phi_c(\boldsymbol{o}_1, \cdots, \boldsymbol{o}_T, \boldsymbol{x}_{1:T})/Z$. Such a model is theoretically appealing because it can cover the dependency between any pair of $\boldsymbol{o}_{t_1}$ and $\boldsymbol{o}_{t_2}$. However, inference in such a large undirected graphic model is typically intractable. One practical approach is to model the local dependency within a fixed time window, which has been used with the trajectory HMM [48], [49]. Despite the improved performance, the training algorithm is still complex. When such a method is combined with a neural network, both training and generation become complicated due to the inversion of high-dimensional matrices [50].

Another practical approach is to model directed or ancestral dependency, i.e., the dependency of $\boldsymbol{o}_t$ on $\boldsymbol{o}_{1:t-1}$. This dependency is referred to as the AR dependency [51]. A model based on the AR dependency, which is referred to as an AR model, can be generally defined as

$$p(\boldsymbol{o}_{1:T}|\boldsymbol{x}_{1:T};\boldsymbol{\Theta}) = \prod_{t=1}^{T} p(\boldsymbol{o}_t|\boldsymbol{o}_{1:t-1}, \boldsymbol{x}_{1:T};\boldsymbol{\Theta}). \tag{8}$$

Using the AR dependency allows $p(\boldsymbol{o}_{1:T}|\boldsymbol{x}_{1:T})$ to be factorized, which makes the model less complex than undirected graphic models. Such an AR model can be trained in a similar manner to a baseline RMDN except the additional cost to load $\boldsymbol{o}_{1:t-1}$ as the additional condition. In the generation stage, $\widehat{\boldsymbol{o}}_{1:T}$ can be generated sequentially. Note that the above definition is also applicable to modeling quantized (or discrete) data.

D. Related AR models

The AR dependency has been widely used in many research fields. One example is the linear predictive coding (LPC) for speech [52], which uses a linear function to model the AR dependency of a speech waveform. For SPSS, the AR-HMM uses linear functions to model the AR dependency of acoustic features [53]. Recently, many models have been defined to model the AR dependency through non-linear transformation. These models are called AR models because the definition of AR in the machine learning field is broader than that in the conventional speech processing field. Examples include the WaveNet [45], variational RNN [54], and RNN language model [55]. The AR dependency can also be defined among latent variables in a neural AR auto-encoder [56].

More generally, the AR dependency is one type of causal dependency defined among random variables. In image processing, the causal dependency can be defined and learned among pixels [57], [58], [59]. For SPSS, a similar idea has

been proposed to model the causal dependency of Mel-cepstral coefficients across dimensions [60].

III. Shallow AR neural F0 model

An AR neural model for continuous data $\boldsymbol{o}_{1:T} \in \mathbb{R}^{T\times D}$ can be implemented on the basis of an RMDN [20]. It defines

$$p(\boldsymbol{o}_{1:T}|\boldsymbol{x}_{1:T};\boldsymbol{\Theta},\boldsymbol{\Psi}) = \prod_{t=1}^{T} p(\boldsymbol{o}_t|\boldsymbol{o}_{t-K:t-1},\boldsymbol{x}_{1:T};\boldsymbol{\Theta},\boldsymbol{\Psi}) = \prod_{t=1}^{T}\sum_{m=1}^{M} \omega_t^m \mathcal{N}(\boldsymbol{o}_t;\boldsymbol{\mu}_t^m + \mathcal{F}_{\boldsymbol{\Psi}}(\boldsymbol{o}_{t-K:t-1}),\boldsymbol{\Sigma}_t^m) \quad (9)$$

where the GMM's parameter set is still given by an internal RNN, i.e., $\boldsymbol{\mathcal{M}}_t = \boldsymbol{\mathcal{H}}_{\boldsymbol{\Theta}}(\boldsymbol{x}_{1:T}, t)$. However, the model uses a function $\mathcal{F}_{\boldsymbol{\Psi}}(\cdot)$ to merge $\boldsymbol{o}_{t-K:t-1}$ in the previous K frames then adjust the mean of each GMM mixture at time t. Specifically, $\mathcal{F}_{\boldsymbol{\Psi}}(\cdot)$ is defined as

$$\mathcal{F}_{\boldsymbol{\Psi}}(\boldsymbol{o}_{t-K:t-1}) = \sum_{k=1}^{K} \boldsymbol{a}_k \odot \boldsymbol{o}_{t-k} + \boldsymbol{b}, \quad (10)$$

where $\boldsymbol{\Psi} = \{\boldsymbol{a}_1, \cdots, \boldsymbol{a}_K, \boldsymbol{b}\}$ is the parameter set that can be jointly trained with $\boldsymbol{\Theta}$, and $\odot$ is the element-wise product. Obviously, $\mathcal{F}_{\boldsymbol{\Psi}}(\cdot)$ makes the distribution of $\boldsymbol{o}_t$ dependent on $\boldsymbol{o}_{t-K:t-1}$. Because $\mathcal{F}_{\boldsymbol{\Psi}}(\cdot)$ is linear and K is finite, this model is referred to as the *shallow AR model* (SAR). The SAR with $K = 2$ is plotted in the middle of Figure 1.

This SAR can be interpreted from the perspective of digital filter and signal. Suppose $\boldsymbol{o}_t \in \mathbb{R}^D$ and the GMM has one mixture component. Then, the distribution of $\boldsymbol{o}_t$ can be written as

$$p(\boldsymbol{o}_t|\boldsymbol{o}_{t-K:t-1},\boldsymbol{x}_{1:T}) = \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi\sigma_{t,d}^2}} \exp\left[-\frac{(o_{t,d} - \sum_{k=1}^{K} a_{k,d} o_{t-k,d} - \mu_{t,d} - b_d)^2}{2\sigma_{t,d}^2}\right], \quad (11)$$

where $o_{t,d}, \mu_{t,d}, a_{k,d}$, and b_d are the d-th dimensions of $\boldsymbol{o}_t, \boldsymbol{\mu}_t$, $\boldsymbol{a}_k$, and $\boldsymbol{b}$, respectively, and $\sigma_{t,d}$ is the d-th diagonal element of the diagonal matrix $\boldsymbol{\Sigma}_t$. Suppose a new random variable is defined as $c_{t,d} = o_{t,d} - \sum_{k=1}^{K} a_{k,d} o_{t-k,d}$, then it can be shown that the vector $\boldsymbol{c}_{1:T,d} = [c_{1,d}, \cdots, c_{T,d}]^\top$ and another vector $\boldsymbol{o}_{1:T,d} = [o_{1,d}, \cdots, o_{T,d}]^\top$ are related by a linear transformation

$$\boldsymbol{c}_{1:T,d} = \boldsymbol{A}^{(d)} \boldsymbol{o}_{1:T,d}, \quad (12)$$

where $\boldsymbol{A}^{(d)} =$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & \cdots & 0 \\ -a_{1,d} & 1 & 0 & 0 & \cdots & 0 & \cdots & 0 \\ -a_{2,d} & -a_{1,d} & 1 & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -a_{K,d} & \cdots & -a_{2,d} & -a_{1,d} & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & -a_{K,d} & \cdots & -a_{2,d} & -a_{1,d} & 1 \end{bmatrix}.$$

Interestingly, Equation (12) is also a filtering process in which the input signal $\boldsymbol{o}_{1:T,d}$ of length T is converted into the output signal $\boldsymbol{c}_{1:T,d}$ by using a finite impulse response (FIR) filter. This FIR filter can be written in the z-domain as $A^d(z) = 1 - \sum_{k=1}^{K} a_{k,d} z^{-k}$, where $\{a_{1,d}, \cdots, a_{K,d}\}$ are the filter coefficients. Because $\boldsymbol{A}^{(d)}$ is invertible, an 'inverse' filtering process can be defined as $\boldsymbol{o}_{1:T,d} = \boldsymbol{H}^{(d)} \boldsymbol{c}_{1:T,d}$, where $\boldsymbol{H}^{(d)} = (\boldsymbol{A}^{(d)})^{-1}$. This 'inverse' filter written in the z-domain is $H^d(z) = 1/A^d(z)$.

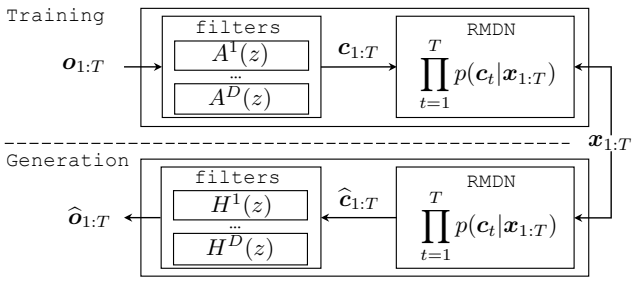


Fig. 2. Conceptual illustration of SAR as RMDN plus digital filters. Note that this SAR is implemented as a single network, as shown in Figure 1.

Generally, $\{H^d(z), A^d(z)\}$ and $\boldsymbol{c}_{1:T,d}$ can be defined for each $d \in [1, D]$. By replacing $o_{t,d} - \sum_{k=1}^{K} a_{k,d} o_{t-k,d}$ in Equation (11) with $c_{t,d}$, it can be demonstrated that

$$\prod_{t=1}^{T} p(\boldsymbol{o}_t|\boldsymbol{o}_{t-K:t-1},\boldsymbol{x}_{1:T}) = \prod_{t=1}^{T} p(\boldsymbol{c}_t|\boldsymbol{x}_{1:T}). \quad (13)$$

Equations (12) and (13) indicate that the SAR can interpreted as a combination of digital filters and an RMDN. This interpretation is shown in Figure 2.

IV. Deep AR neural F0 model

A. Weakness of SAR and RMDN

The local and linear AR dependency modeled in the SAR may be insufficient for F0 modeling. This weakness can be revealed by examining the F0 contours sampled from the model [61]. Let's use the SAR in Section V as the example. Given this SAR, $\widehat{\boldsymbol{o}}_1$ can be sampled from $p(\boldsymbol{o}_1|\boldsymbol{x}_{1:T})$, and then $\widehat{\boldsymbol{o}}_2$ can be drawn from $p(\boldsymbol{o}_2|\widehat{\boldsymbol{o}}_1, \boldsymbol{x}_{1:T})$. Repeating this ancestral sampling process can generate an F0 contour. If the AR dependency is strong enough, $\widehat{\boldsymbol{o}}_{t+1}$ and $\widehat{\boldsymbol{o}}_t$ are likely to have a similar value, and the sampled contour should be as smooth as a natural F0 contour. However, Figure 3 shows that the sampled F0 from the SAR is very noisy.

For reference, Figure 3 shows an F0 contour sampled from a baseline RMDN, which is also trained on continuous F0 data and used in Section V. The noisy output from this RMDN is expected because F0 values are drawn from independent distributions. In the case of the SAR, sampling an F0 contour is equivalent to drawing an F0 contour from a virtual RMDN and filtering it. However, the linear filter in this SAR seems to be incapable of removing the rapid change in the sampled F0 contour. We observed that this weakness remains even if the order K of the filter is increased. Note that the SAR and the RMDN can generate smooth contours by using the mean-based generation method, which is shown later in Figure 9 and explained in the experiments of Section V-E. However, it only indicates the smoothness of the F0 mean trajectory but not necessarily the goodness of the model.

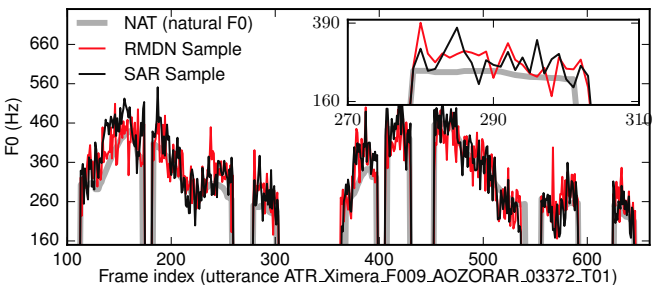


Fig. 3. F0 contours sampled from RMDN and SAR. The model configuration is explained in Section V-B. The embedded figure shows the details around the 290th frame. Note that the variance of distribution was scaled by 0.5 before sampling. Otherwise, sampled F0 contours were too noisy for visualization.

B. Definition of deep AR model

A better implementation for the AR dependency may require non-linear transformations. This study followed the approach to feed the target data of the previous frame as the additional input to a uni-directional recurrent layer at the current frame [43]. The implementation is straightforward: concatenate $\boldsymbol{o}_{t-1}$ with the output of the previous hidden layer at the t-th frame and use the concatenated vector as an input to the recurrent layer. While the natural $\boldsymbol{o}_{t-1}$ is fed back during model training, the previously generated $\widehat{\boldsymbol{o}}_{t-1}$ or other statistics introduced in Section IV-D can be fed back during generation.

Because the feedback data are propagated by the recurrent layer, hidden features extracted from $\boldsymbol{o}_{1:t-1}$ can be used at the t-th frame. Therefore, the output of the internal RNN at the t-th frame, or consequently the parameter $\boldsymbol{\mathcal{M}}_t$ of $\boldsymbol{o}_t$'s distribution, can be computed as $\boldsymbol{\mathcal{M}}_t = \boldsymbol{\mathcal{H}}_{\boldsymbol{\Theta}}(\boldsymbol{x}_{1:T}, \boldsymbol{o}_{1:t-1}, t)$. Consequently, the distribution of $\boldsymbol{o}_t$ depends on $\boldsymbol{o}_{1:t-1}$ in the same manner as Equation (8) shows. Since the AR dependency is non-linear and potentially beyond a local time window, this proposed model is referred to as the *deep AR* model (DAR). The data-feedback path is referred to as the *feedback link*. One example of the DAR is plotted on the right side of Figure 1.

C. Comparing DAR and SAR with RMDN and its extensions

The proposed DAR and the SAR only introduce additional recurrent or feedback links compared to an RMDN. Some readers may wonder what the differences are between AR models and extended RMDNs with similar links inside the internal RNN. Such a toy extended RMDN with a recurrent link in the output layer [62] is shown on the left side of Figure 4. One difference is that, while the link in the SAR and DAR delivers $\boldsymbol{o}_t$, the link in the extended RMDN carries the valued computed from the internal RNN. This difference impacts the model's capability to learn the temporal dependency of $\boldsymbol{o}_{1:T}$.

Let us explain this difference intuitively using the toy models in Figure 4, where the target data sequence is $\boldsymbol{o}_{1:2} = [o_1, o_2]$ and $o_1, o_2 \in \mathbb{R}$. To simplify the explanation, we assume that all the layers use a linear activation function and set the bias to zero. The distribution for o_t is assumed to be a Gaussian distribution with a unit variance.

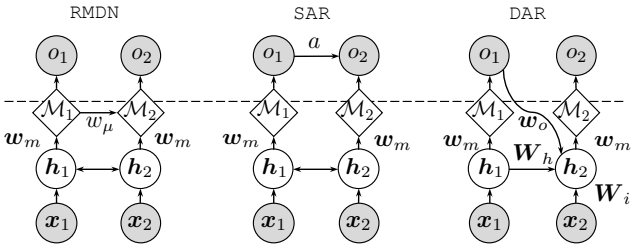


Fig. 4. Illustration of toy neural models. The RMDN uses a recurrent output layer [62]. $\boldsymbol{w}_*$ and $\boldsymbol{W}_*$ denote the transformation vectors and matrices.

a) Extended RMDN: For the toy extended RMDN, it is easy to show that $\mathcal{M}_1 \triangleq \mu_1 = \boldsymbol{w}_m^\top \boldsymbol{h}_1$ and $\mathcal{M}_2 \triangleq \mu_2 = \boldsymbol{w}_m^\top \boldsymbol{h}_2 + w_\mu \mu_1$. Let $\tilde{\mu}_2 = \boldsymbol{w}_m^\top \boldsymbol{h}_2$, then the conditional distribution of $\boldsymbol{o}_{1:2}$ can be written as

$$\begin{aligned} p(o_{1:2}|\boldsymbol{x}_{1:2}) =& \mathcal{N}(o_1; \mu_1, 1)\mathcal{N}(o_2; \tilde{\mu}_2 + w_\mu \mu_1, 1) \\ =& \frac{1}{2\pi} \exp(-\frac{(o_1-\mu_1)^2}{2} - \frac{(o_2 - \tilde{\mu}_2 - w_\mu \mu_1)^2}{2}) \\ =& \frac{1}{2\pi} \exp(-\frac{1}{2}(\boldsymbol{o}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{o}-\boldsymbol{\mu})), \end{aligned} \tag{14}$$

where $\boldsymbol{o} = [o_1, o_2]^\top$, $\boldsymbol{\mu} = [\mu_1, \tilde{\mu}_2 + w_\mu \mu_1]^\top$, and $\boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Despite the recurrent link in the output layer, the covariance matrix $\boldsymbol{\Sigma}$ is diagonal, which suggests that o_1 and o_2 are treated with the model as independent random variables.

b) SAR: On the other hand, the toy SAR computes $\mathcal{M}_1 \triangleq \mu_1 = \boldsymbol{w}_m^\top \boldsymbol{h}_1$ and $\mathcal{M}_2 \triangleq \mu_2 = \boldsymbol{w}_m^\top \boldsymbol{h}_2$. Accordingly, the distribution calculated with the model can be written as

$$\begin{aligned} p(o_{1:2}|\boldsymbol{x}_{1:2}) =& \mathcal{N}(o_1; \mu_1, 1)\mathcal{N}(o_2; \mu_2 + a o_1, 1) \\ =& \frac{1}{2\pi} \exp(-\frac{(o_1-\mu_1)^2}{2} - \frac{(o_2 - \mu_2 - a o_1)^2}{2}) \\ =& \frac{1}{2\pi} \exp(-\frac{1}{2}(\boldsymbol{o}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{o}-\boldsymbol{\mu})), \end{aligned} \tag{15}$$

where $\boldsymbol{o} = [o_1, o_2]^\top$, $\boldsymbol{\mu} = [\mu_1, \mu_2 + a\mu_1]^\top$, $\boldsymbol{\Sigma} = \begin{bmatrix} 1 & a \\ a & 1+a^2 \end{bmatrix}$, and a is the trainable AR parameter. As long as $a \neq 0$, the covariance matrix $\boldsymbol{\Sigma}$ becomes a full matrix, which means that the correlation between o_1 and o_2 is not ignored with the model.

c) DAR: Similarly, the toy DAR computes $\mu_1 = \boldsymbol{w}_m^\top \boldsymbol{h}_1$ and $\mu_2 = \boldsymbol{w}_m^\top \boldsymbol{h}_2$, where $\boldsymbol{h}_2 = \boldsymbol{W}_h \boldsymbol{h}_1 + \boldsymbol{W}_i \boldsymbol{x}_2 + \boldsymbol{w}_o o_1$. If we define $\tilde{\mu}_2 = \boldsymbol{w}_m^\top (\boldsymbol{W}_h \boldsymbol{h}_1 + \boldsymbol{W}_i \boldsymbol{x}_2)$, then we can get

$$p(o_{1:2}|\boldsymbol{x}_{1:2}) = \mathcal{N}(o_1; \mu_1, 1)\mathcal{N}(o_2; \tilde{\mu}_2 + \boldsymbol{w}_m^\top \boldsymbol{w}_o o_1, 1). \tag{16}$$

This equation indicates that the toy DAR models the dependency between o_1 and o_2 in a similar manner to the toy SAR.

As the toy examples suggest, AR models differ from baseline models because the recurrent or feedback link delivers the target data $\boldsymbol{o}_t$ rather than output of the internal RNN. Compared with the SAR, the DAR may be more general. When the toy DAR uses a non-linear activation function $\sigma(\cdot)$, it computes $\boldsymbol{h}_2 = \sigma(\boldsymbol{W}_h \boldsymbol{h}_1 + \boldsymbol{W}_i \boldsymbol{x}_2 + \boldsymbol{w}_o o_1)$. Then, the mean of o_2's distribution becomes a non-linear function of o_1. Furthermore, if $\boldsymbol{o}_{1:T}$ has more than two frames, the

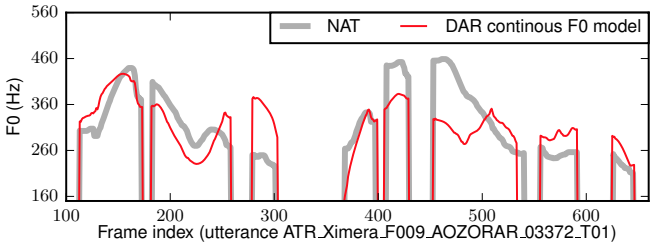


Fig. 5. Generated F0 contour from DAR-based continuous F0 model by using mean-based generation. NAT denotes natural F0.

distribution of o_t is also affected by $\boldsymbol{o}_{1:t-2}$ because hidden features extracted from $\boldsymbol{o}_{1:t-2}$ are delivered by $\boldsymbol{h}_{t-1}$. Note that the advantage of the DAR is valid even if it is used for modeling quantized or discrete data.

D. DAR for quantized F0 modeling

A DAR-based F0 model seems promising, but a practical issue on the representation of F0 data has to be solved. Because the F0 of an unvoiced frame is unmeasurable, an F0 datum is defined as $o_t \in \{\text{NULL}\} \cup \mathbb{R}$ with the classical HMM-based SPSS framework, where NULL is the symbol for the unvoiced frame. These F0 data can be modeled using a multi-space probability distribution HMM [37]. Alternatively, artificial F0 values can be assigned to the unvoiced frames, after which the continuous F0 and unvoiced/voiced (U/V) state of each frame can be modeled using a normal HMM [9]. Note that 'continuous' means that $o_t \in \mathbb{R}, \forall t \in \{1, \cdots, T\}$. This continuous F0 modeling approach is widely used with NN-based F0 models including the baseline RNN [19] and the SAR. However, our experiments showed that the DAR performed poorly on continuous F0 data. After training the DAR, the configuration of which was based on RMDN in Section V, the correlation between generated and natural F0 significantly dropped from 0.89 (RMDN's score) to 0.58 on the test set. A generated F0 contour is plotted in Figure 5. We strongly believe that this degradation is due to the artificial F0 values interpolated in the unvoiced frames. Since they are artificially interpolated, they have deterministic dependency to previous frames, whereas the natural F0 values in the voiced frames have stochastic dependency. This may disturb the learning of temporal dependency in natural F0 contours.

Hence, the DAR requires an alternative method to represent both unvoiced and voiced frames without using F0 interpolation. We propose a strategy to quantize the F0 value of voiced frames and further assign one additional class to unvoiced frames. In this manner, both voiced and unvoiced frames can be represented as a set of categorical symbols. This strategy is also reasonable because of the just-noticeable difference of pitch in speech perception, as Section II-B discussed. To quantize the F0 data, the first step is to map the original F0 onto a Mel scale, then the Mel-scale F0 is quantized into N levels. Finally, the quantized F0 of one frame can be encoded as a one-hot vector $\boldsymbol{o}_t = [o_{t,0}, o_{t,1}, \cdots, o_{t,N}]$, where $o_{t,j} \in \{0, 1\}$ and $||\boldsymbol{o}_t||_1 = 1$. If the frame is unvoiced, the 1st dimension of this one-hot vector is set to one, i.e., $\boldsymbol{o}_t = [1, 0, \cdots, 0]$.

After F0 quantization, the F0 contour becomes a sequence of categorical symbols. This representation is very convenient since it allows F0 values and U/V state to be modeled simultaneously. The DAR may use a softmax output layer to calculate a probability for each of the quantized F0 and unvoiced symbol at each frame. However, from the experiments discussed in Section V-C, a normal softmax layer is not the best choice because the quantity of unvoiced data in the corpus is much larger than that of any other quantized F0 symbol. We thus suggest using a hierarchical softmax layer [63] to deal with the unbalanced data distribution. Suppose that the hot dimension of $\boldsymbol{o}_t$ is indexed by j, then the PMF w.r.t. $\boldsymbol{o}_t$ can be defined as $P(\boldsymbol{o}_t|\boldsymbol{o}_{1:t-1}, \boldsymbol{x}_{1:T}; \boldsymbol{\Theta}) \triangleq P_t(J = j)$, where

$$P_t(J=j) = \begin{cases} \dfrac{e^{h_{t,0}}}{1+e^{h_{t,0}}}, & j \in \{0\} \\ \dfrac{1}{1+e^{h_{t,0}}} \dfrac{e^{h_{t,j}}}{\sum_{k=1}^{N} e^{h_{t,k}}}, & j \in [1, N] \end{cases}. \quad (17)$$

Here, $h_{t,j}$ is the j-th dimension of the input vector $\boldsymbol{h}_t$ to the hierarchical softmax layer. This $\boldsymbol{h}_t$ is calculated from the network given feedback data $\boldsymbol{o}_{1:t-1}$ and linguistic features $\boldsymbol{x}_{1:T}$. Accordingly, it can be written that $P_t(J = j) = \boldsymbol{\mathcal{H}}_{\boldsymbol{\Theta}}(\boldsymbol{x}_{1:T}, \boldsymbol{o}_{1:t-1}, t, j)$.

As Equation (17) shows, the first hierarchical level uses a sigmoid function to compute a probability of being unvoiced, i.e., $P_t(\text{unvoiced}) \triangleq P_t(J = 0) = \frac{e^{h_{t,0}}}{1+e^{h_{t,0}}}$. The second level uses a normal softmax function to compute a conditional probability of each quantized F0 symbol given a voiced state, i.e., $P_t(J = j|\text{voiced}) = \frac{e^{h_{t,j}}}{\sum_{k=1}^{N} e^{h_{t,k}}}, j \in [1, N]$. Based on Equation (17), the network can be trained by maximizing the likelihood or equivalently minimizing the cross-entropy.

In the generation stage, if $P_t(J = 0) > 0.5$, the t-th frame is classified as being unvoiced. Otherwise, this frame will be voiced and will be assigned a real-number F0 value $\widehat{f}_t$. Suppose $\{v_1, \cdots, v_N\}$ denotes the real F0 values of the N quantization levels, e.g., the center of each quantization interval. The F0 value $\widehat{f}_t$ can be acquired using a random-sampling-based generation method, which is written as

$$\widehat{f}_t = v_j, \text{where } j \sim P_t(J = j|\text{voiced}), j \in \{1, \cdots, N\}. \quad (18)$$

Given the sampled value j, a one-hot vector $\widehat{\boldsymbol{o}}_t$ with the j-th dimension turned on is fed back to the next frame. Remember that random-sampling is used to test the model's capability.

So far, we treated the F0 symbols as discrete categorical values. Since the F0 symbols for voiced frames are quantized F0 indexes, a real-number F0 value $\widehat{f}_t$ can also be generated from expected value below

$$\widehat{f}_t = \sum_{j=1}^{N} v_j P_t(J = j|\text{voiced}). \quad (19)$$

This method is the *mean-based generation* method for quantized F0. After $\widehat{f}_t$ is generated, a vector of probability $[P_t(J = 0), P_t(J = 1), \cdots, P_t(J = N)]$ is fed back to the next frame.

E. Exposure bias of DAR

The representation of F0 data is not the only issue that affects the DAR. As Section IV-B explains, the feedback link in the DAR propagates the natural F0 during model training, which is known as teacher-forced training [64]. However, because the natural $\boldsymbol{o}_{t-1}$ usually has a similar value to $\boldsymbol{o}_t$, a trained DAR may only rely on the feedback F0 data while ignoring the input linguistic features $\boldsymbol{x}_{1:T}$. This behavior is unwanted because an F0 model should also use the linguistic features. Furthermore, while natural F0 data are propagated through the feedback links in the training stage, generated F0 data are fed back during generation. Because the distribution of generated F0 may not be identical to the natural one, the model may suffer from the problem called *exposure bias* [65]. Consequently, generation errors in the previous frames may be propagated to the next frame, which makes the entire generated F0 contour erratic.

In this study, we propose a *data dropout* strategy to alleviate the above problems. It requires the model to randomly set the feedback F0 data to zero in both training and generation stages, which consequently forces the DAR to focus on the linguistic features. This strategy is similar to the idea of weakening the AR-model-based decoder for lossy variational auto-encoders [66]. It will also alleviate the exposure bias as the model relies more on the linguistic features that are given with the same TTS front-end for both training and generation.

The problem of exposure bias has been investigated for the task of structured prediction, and several methods, such as data as demonstrator (DAD) [67] and scheduled sampling [68], have been proposed. Their basic strategy is to feed back the generated data during training. However, it is also known that this strategy may force the model to ignore the temporal dependency of the natural data sequence [69]. An alternative method is to combine search and optimization for model training [70]. The idea is to train a model by minimizing the average utterance-level distance between natural and generated data sequence candidates, which does not require feeding back the natural data during training. We leave this idea for our future work because of increased complexity in pruning and searching the space of generated data sequences.

V. Experiments

A. Corpus and feature extraction

We conducted experiments on a Japanese speech corpus, which was used for the XIMERA unit-selection TTS system [71]. All the data are neutral, read speech recorded at a sampling rate of 48 kHz. Fifty hours of recordings from a female speaker (F009) were used for the experiments. This subset contained 30,016 segmented utterances, among which 500 were randomly selected as the validation set and another 500 as the test set. Note that the validation and test sets were larger than those used in a previous experiment [72].

Linguistic features were automatically extracted from the speech transcription using the TTS front-end called OpenJTalk [73]. This front-end conducted grapheme-to-phoneme conversion, part-of-speech tagging, and syntactic parsing based on the Mecab toolkit [74]. The output from the front-end was identical to that used in the Japanese HTS system [75], including quin-phone identity, word part-of-speech tag, phrase accent type, and other structural information. These linguistic features were encoded into a vector of 389 dimensions.

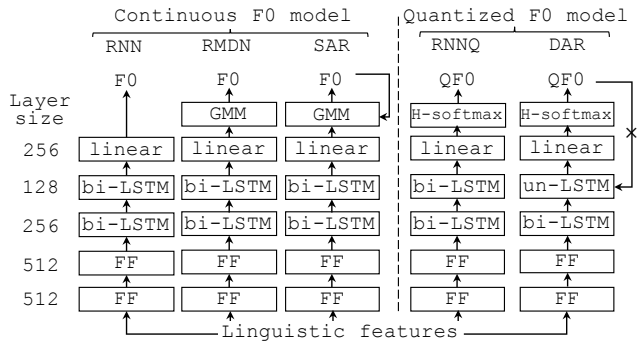


Fig. 6. Structure of experimental models. FF, H-softmax, bi-LSTM, and un-LSTM denote feedforward, hierarchical-softmax, bi-directional, and uni-directional LSTM layer, respectively. QF0 denotes quantized F0 data. $\times$ in DAR means random data dropout. Layer size of bi-LSTM is the sum of the forward and backward recurrent layers' size.

Natural F0 values were extracted by merging the results of multiple pitch trackers [76] with the frame rate of 200 Hz (5 ms). Then, these raw F0 data were transformed to the Mel scale [77] by using $m = 1127 \log(1 + \mathrm{F0}/700)$. For the models working on continuous F0, the unvoiced frames were interpolated using an exponential function. For quantized F0 models, the transformed F0 values were quantized into 255 levels between 66 and 529 on the Mel scale. The number 66 was equal to the minimum Mel-scale F0 value in the corpus. The number 529 was computed with $m+3\sigma$, where $m = 342.4$ and $\sigma = 62.2$ are the mean and standard deviation of Mel-scale F0 over the corpus. The number of quantization levels was decided from an analysis-by-synthesis test, which found that using 255 levels was sufficient to avoid F0 'quantization noise'. The quantized F0 and unvoiced symbol were encoded as a one-hot vector $\boldsymbol{o}_t \in \{0, 1\}^{256}$ for each frame. The F0 delta and delta-delta components were not used.

Speech samples for the listening test were generated given natural Mel-generalized cepstral coefficients [78] of order 60 and band aperiodicity coefficients of order 25. The WORLD vocoder [79] was used for waveform generation from the acoustic features.

B. Model configuration, training, and testing

The experimental models are shown in Figure 6, among which RNN, RMDN and SAR work on the continuous F0 while the RNN-based quantized F0 model (RNNQ) and the proposed DAR work on the quantized F0. Note that RNNQ is just the baseline quantized F0 model defined in Equation (7).

After the input layer, all the models used two feedforward layers with 512 nodes and a bi-directional LSTM layer with 256 nodes. The feedforward layers used the $tanh$-based activation function. After the first three hidden layers, all models, except DAR, used another bi-directional LSTM layer with 128 nodes. On the output side, RNN used a linear output layer. RMDN and SAR used a linear layer to generate the parameters

for a binomial distribution modeling U/V status and a GMM with two mixture components modeling the F0 value. The motivation to use two mixture components instead of one was to make the model robust against data outliers. SAR set $K = 1$ for the AR dependency. This choice of K was based on previous experiments in which a larger K did not improve performance. Different from other models, DAR used a uni-directional LSTM of 128 nodes after the first three hidden layers. This uni-LSTM layer also took the feedback F0 as the input. After this LSTM layer, a linear layer was used to generate the activation to the hierarchical softmax layer. Several versions of DAR were trained and tested using different dropout rates $P_d = \{0.00, 0.25, 0.50, 0.75\}$.

The plain stochastic gradient descent (SGD) with early stopping was used for initial training. After the error on the validation set consistently increased for five epochs, the best trained model was further tuned using the AdaGrad optimizer [80] with early stopping. The learning-rate for the SGD was 1e-5, while the learning-rate parameter for AdaGrad was 0.001. RNN, RNNQ, and DAR were initialized using the layer-size-dependent uniform distribution [81]. RMDN was initialized with the trained RNN except the last linear layer. Similarly, SAR was initialized by the trained RMDN. All models and experiments were implemented on the basis of the CURRENNT toolkit [82]. The toolkit and training recipes for the experiments are publicly available online (http://tonywangx.github.io)

To evaluate the performance of the experimental models, the log-likelihood of each model except RNN was calculated on the validation set. Natural F0 data were fed back when the log-likelihood of DAR and SAR was evaluated. For further evaluation, subjective and objective tests were conducted on the test set. The subjective test is discussed in Section V-E. For the objective test, several metrics were calculated on generated F0 given the natural duration information. These metrics include root mean square error (**RMSE**), a correlation coefficient (**CORR**), a U/V classification error rate (**U/V**), and utterance-level F0 global variance (**GV**). The **RMSE** and **CORR** were calculated on frames where both natural and generated F0s were voiced. For different versions of DAR using random dropout ($P_d = \{0.25, 0.50, 0.75\}$), the objective evaluation was conducted for 10 rounds, where the model used a different random seed for F0 generation in each round. The mean value of each metric over the 10 rounds is reported in this paper. Standard deviation is not shown because it is quite small: $\sigma_{\text{RMSE}} < 0.2$, $\sigma_{\text{CORR}} < 0.0017$, $\sigma_{\text{UV}} < 0.018\%$, and $\sigma_{\text{GV}} < 0.5$ for all related models.

C. Effectiveness of hierarchical softmax for DAR

Before evaluating DAR against other models, this section discusses the effectiveness of hierarchical softmax. This test introduced another DAR-based model that had the same structure as DAR except using a normal softmax layer. This model was trained using dropout rates $P_d = \{0.00, 0.25, 0.50, 0.75\}$. Note that this model generates an unvoiced frame if $P_t(\text{unvoiced})$ is larger than the probability of other F0 symbols. Otherwise, it uses the mean-based generation method in Equation (19) to generate F0. The objective evaluation was conducted for 10 rounds when $P_d = \{0.25, 0.50, 0.75\}$.

TABLE I
PERFORMANCE OF DAR USING DIFFERENT SOFTMAX LAYERS. MEAN-BASED GENERATION WAS USED. **U→V** DENOTES ERROR RATE OF CLASSIFYING UNVOICED FRAMES AS VOICED; **V→U** DENOTES THE ERROR THE OTHER WAY ROUND.

	Softmax type	U/V	V→U	U→V
$P_d = 0.75$	Normal	5.31%	4.57%	0.74%
	Hierarchical	3.35%	1.66%	1.69%
$P_d = 0.50$	Normal	4.87%	4.04%	0.83%
	Hierarchical	3.46%	1.86%	1.60%
$P_d = 0.25$	Normal	4.55%	3.46%	1.09%
	Hierarchical	3.62%	1.86%	1.76%
$P_d = 0.00$	Normal	4.43%	2.98%	1.45%
	Hierarchical	3.82%	1.73%	2.09%

No matter what the dropout rate was, using the normal softmax layer achieved similar but no better RMSE and CORR scores than the hierarchical one. The results of RMSE and CORR are thus not shown in this paper. What is interesting is the results on U/V decision. Table I lists the gross U/V error rate, the error rate of classifying voiced frames as unvoiced (V→U), and the error rate the other way around (U→V). Interestingly, while using the normal softmax layer achieved a lower U→V error rate, it made more V→U errors. The unbalanced error rate was not observed in the case of the hierarchical softmax. In terms of gross error rate, using the hierarchical softmax layer performed better. Similar results were also observed from an experiment on English data [21].

These results may be due to the imbalance of F0 data distribution. While around 50% of training data are unvoiced (including 28% silent frames), the ratio of any quantized F0 symbol is less than 0.6%. Therefore, the model with a normal softmax layer may be trained to over-estimate the probability for being unvoiced, which may have caused a low U→V error rate but a high V→U error rate. For the model using the hierarchical softmax layer, the U/V decision is a binary classification problem and can be learned given well balanced U/V data.

D. Effectiveness of dropout for DAR

This section analyzes the usefulness of dropout on DAR. The F0 contours on the test set were generated using the mean-based generation method for discrete data (Equation (19)). Note that RNNQ can be treated as DAR with $P_d = 1.0$. The results in Table II indicate that DAR with $P_d > 0$ acquired better RMSE and CORR scores than DAR with $P_d = 0.0$, which indicates the effectiveness of using dropout on DAR.

As Section IV-E argues, a potential problem of using DAR is the exposure bias. This is supported from the results in Table II. First, DAR with $P_d = 0.0$ achieved a higher likelihood than other cases on the development set. Because the likelihood on the development set was evaluated given the natural F0 data for feedback, it suggests that, when the natural F0 data were used as the condition, DAR could better depict the F0 data without using data dropout. However, on the test set in which the model

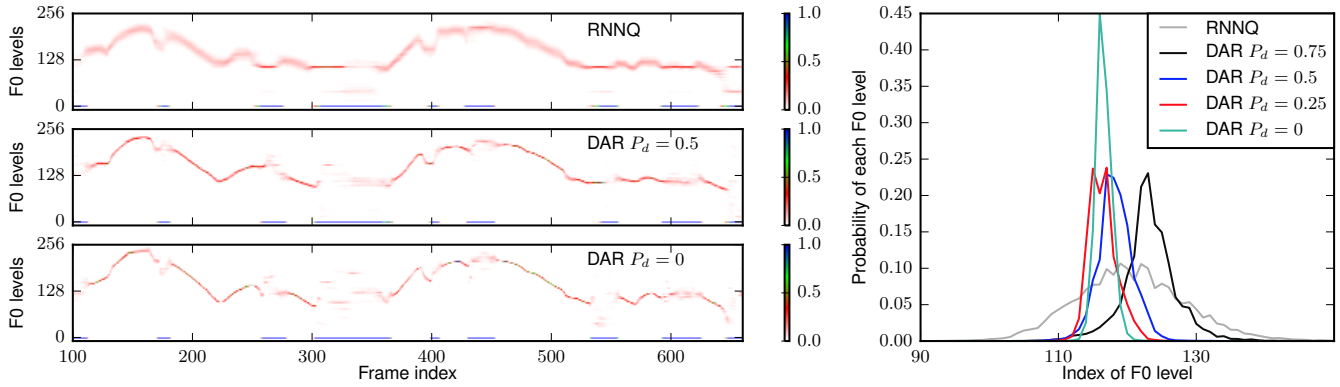


Fig. 7. Left: inferred F0 probability ($P(\boldsymbol{o}_t|\boldsymbol{h}_t)$ in Equation (17)) for one test utterance. Right: inferred F0 probability for the 200th frame of same test utterance. Note that the 0th F0 level denotes state of 'being unvoiced' while other levels denote quantized F0 levels.

TABLE II
RESULTS OF OBJECTIVE EVALUATION ON TEST SET. RNN, RMDN, AND SAR USED MEAN-BASED GENERATION METHOD FOR CONTINUOUS F0; DAR AND RNNQ USED MEAN-BASED GENERATION METHOD FOR QUANTIZED F0. P_d DENOTES DROPOUT PROBABILITY; GV OF NATURAL F0 IS AROUND 61.

	P_d	Log-likelihood validation set	RMSE	CORR	U/V	GV
RNN	-	-	29.31	0.894	3.26%	51.4
RMDN	-	-3340.9	28.32	0.897	3.85%	54.2
SAR	-	-3278.6	34.57	0.898	3.85%	71.8
RNNQ	-	-2443.8	26.77	0.907	5.74%	56.4
DAR	0.75	-1595.1	26.52	0.909	3.35%	57.8
	0.50	-1156.3	28.30	0.903	3.46%	61.5
	0.25	-943.6	29.70	0.896	3.62%	62.7
	0.00	-839.7	32.04	0.881	3.82%	64.4

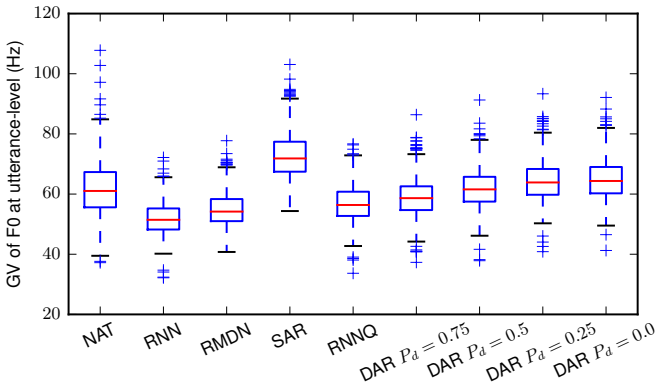


Fig. 8. Box-plot of GV for natural and generated F0 on test set

had to feed back the generated F0 data, DAR's performance in terms of RMSE and CORR degraded. In fact, DAR with $P_d = 0.0$ was very confident in terms of the F0 distribution. This confidence can be demonstrated by the small variance of its F0 distribution in Figure 7. However, a small variance does not necessarily mean a small bias. When DAR uses its output as feedback data on the test set, a slight difference between the generated and natural F0 may be propagated to the following frames and accumulated.

Dropout may provide DAR with the freedom to adjust the bias and variance. As Table II and Figure 7 show, a larger P_d led to a larger variance of the F0 distribution; consequently, a smaller likelihood on the development set. However, it improved the RMSE and CORR on the test set. Particularly, DAR with $P_d = 0.75$ achieved the best performance in terms of RMSE and CORR.

The rate of dropout should be selected carefully. An appropriate choice should strike a balance between the bias and variance. Another concern indicated in Figure 8 is that intensive dropout may reduce the GV of the generated F0 contours and make them over-smoothed. Because the over-smoothing effect may be more harmful to the perception of pitch than the degraded value of RMSE or CORR, DAR with $P_d = 0.5$ was selected for the subjective evaluation discussed in the next section.

E. Comparison among DAR, SAR, and baseline models

1) Objective evaluation: This section compares the performance of the experimental models. Note that RNN, RMDN, and SAR used the mean-based generation method for continuous F0 while DAR and RNNQ used the mean-based generation method for quantized F0. The objective results are listed in Table II, and the generated F0 contours are plotted in Figure 9.

The results indicate that RNN and RMDN were strong baselines. Compared with RMDN and RNN, SAR performed worse in terms of RMSE and similarly in terms of CORR. However, SAR acquired a higher GV score, and the generated F0 contours had a larger dynamic range. SAR's performance may be explained if SAR is interpreted on the basis of the RMDN-plus-filter model shown in Figure 2. Given the AR coefficients from the trained SAR, the frequency responses of those virtual filters are plotted in Figure 10. Note that the analysis filter $A(z)$ enhances the high-frequency band while suppressing the low-frequency part of the input F0 contour. Because the energy of an F0 contour concentrates in the low-frequency band, what $A(z)$ does is similar to signal whitening. Accordingly, the virtual RMDN in SAR only models the 'whitened' F0 contours, and the averaging effect of statistic modeling may have less impact on the original F0 contours. This may be the reason for the increased GV of SAR. However, it generated less smoothed F0 curves such as the segment around the 500th-frame in Figure 9. These under-smoothed curves degraded the RMSE score and turned out to be perceptible in the subjective

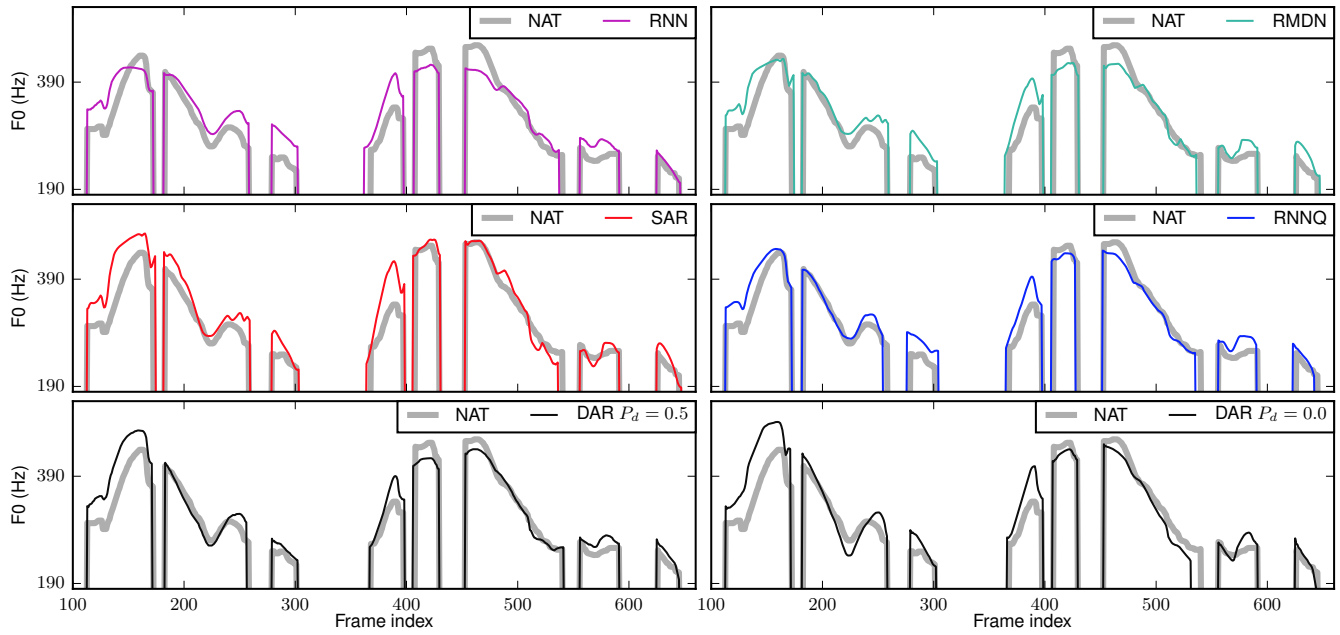


Fig. 9. Generated F0 contours for test utterance (AOZORAR_03372_T01). NAT denotes natural F0. RNN, RMDN, and SAR used the mean-based generation method for continuous F0 data. DAR and RNNQ used the mean-based generation method for quantized F0 data.

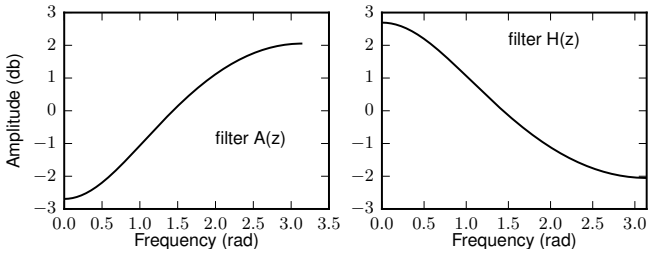
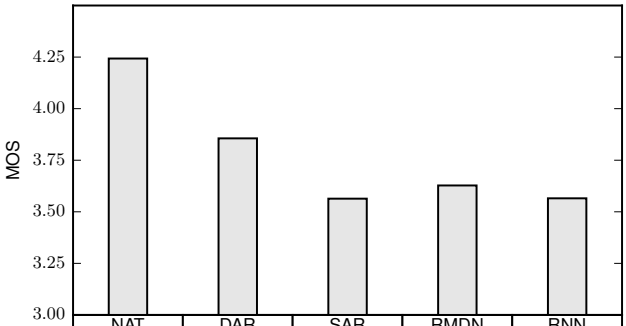


Fig. 10. Frequency response of filters $A(z)$ and $H(z)$ learned with SAR



	NAT	DAR	SAR	RMDN	RNN
NAT		<1e-16	<1e-16	<1e-16	<1e-16
DAR	<1e-16		<1e-16	<1e-16	<1e-16
SAR	<1e-16	<1e-16		0.01785	0.7429
RMDN	<1e-16	<1e-16	0.01785		0.00426
RNN	<1e-16	<1e-16	0.7429	0.00426	

Fig. 11. Results of subjective evaluation. NAT denotes the natural F0. $p-$values calculated from the two-sided Mann-Whitney U test are listed in the table.

evaluation.

Compared with SAR and other baselines, DAR performed better when an appropriate dropout rate was used. For example, DAR with $P_d = 0.75$ achieved the highest CORR and lowest RMSE. DAR with $P_d = 0.5$ also achieved good objective scores. As Figure 9 shows, the generated F0 contours from DAR were sufficiently smooth, even though the models were trained given quantized F0 data. However, these generated F0 contours are not over-smoothed. For example, Table II shows that DAR with $P_d = 0.5$ acquired a GV score that was closer to the natural F0. In particular, from Figure 9, the generated F0 contour from DAR with $P_d = 0.5$ had a larger dynamic range than those from the baseline RNN and RMDN.

2) Subjective evaluation: To further evaluate the performance of the experimental models, a mean-opinion-score (MOS) test was conducted to compare RNN, RMDN, SAR, DAR using $P_d = 0.5$, and natural F0 (NAT). As Section V-A explained, speech samples were generated given natural spectral features. In one evaluation set of the MOS test, each participant had to complete five evaluation screens. On each screen, a sample from one of the five systems was played, and the participant was asked to rate the prosody from "1 - unnatural" to "5 - natural". The five samples in one evaluation set had the same linguistic content, and the order of these samples was randomly shuffled.

This MOS test was crowdsourced online, and 130 paid native Japanese speakers participated. Each participant completed around 23 evaluation sets on average, and 3000 sets of MOS scores were collected in total. The results are plotted in Figure 11. Although it was still worse than NAT, DAR outperformed other models. Results of two-sided Mann-Whitney U tests demonstrated that the difference between DAR and other F0 models was statistically significant ($p < 0.01$). One main reason for DAR's performance may be that the generated F0 contours were sufficiently smooth but not over-smoothed. Compared with SAR, DAR's output did not contain the under-smoothed curves that turned out to be perceptually harmful to the synthetic speech. Meanwhile, generated F0 contours from DAR had a proper dynamic range and sounded less boring than those from the baseline models.

TABLE III
RESULTS OF OBJECTIVE EVALUATION ON SAMPLED F0 CONTOURS FROM DAR ($P_d = 0.5$)

Linguistic features	Sampling rounds	RMSE	CORR	U/V	GV
Full set of linguistic features	1st round	30.49	0.889	3.63%	60.6
	2nd round	30.65	0.887	3.59%	60.8
	3rd round	30.31	0.890	3.65%	60.9
Without pitch accents	1st round	35.93	0.840	3.57%	61.4
	2nd round	36.52	0.838	3.60%	61.9
	3rd round	36.02	0.838	3.63%	61.2

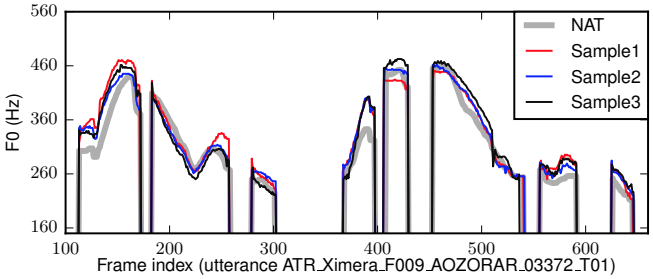
F. Random sampling on DAR

The results given in the previous section indicate DAR's potential for F0 modeling. However, the results did not directly answer whether DAR's performance is due to its improved capability to model sequential data. This section first discusses the test on DAR by using the random-sampling-based generation method. This test was conducted on DAR with $P_d = 0.5$ for three rounds, where each round used a different random seed for sampling. Figure 12 (a) plots the three randomly sampled F0 contours for one test-set sentence. Interestingly, the randomly sampled F0 contours were smooth and much better than the output of RMDN and SAR shown in Figure 3. Furthermore, these sampled F0 contours were quite close to the natural one. The objective metrics calculated on the randomly sampled F0 contours are listed in the first three rows of Table III. Comparison between these results and those in Table II indicates that the random-sampling generation method achieved similar objective scores to the mean-based generation method on DAR. Note that the sampled F0 contours in Figure 12 (a) contained small spikes due to the random sampling process. However, they are barely perceptible. Generally, these results indicate that DAR is better at F0 modeling than SAR and RMDN.

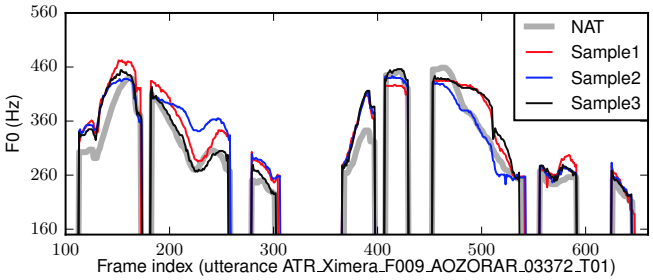
It is not surprising to see that smooth F0 contours can be sampled from DAR. As Figure 7 shows, the F0 distribution calculated with DAR has a sharp mode, and this mode moves slowly across frames. Thus, it is highly possible to sample a smooth F0 contour from the sequence of such F0 distribution. However, it is surprising that the sampled F0 contours were quite similar to the natural one. Despite the detailed differences, the sampled and natural F0 contours were perceived to be quite similar in terms of intonation.

We hypothesized that this is due to the characteristics of Japanese speech data. In the case of reading speech, the F0 contour of an utterance may be sufficiently specified by the Japanese pitch accents. Although the Japanese pitch accents interact with each other in an utterance [83], they can be somewhat pinned down by a lexicon. Therefore, in the TTS system, the input linguistic features given by the front-end may be sufficiently informative for DAR to determine the shape of generated F0 contours; thus, leaving less space for sampling F0 contours with varied shapes.

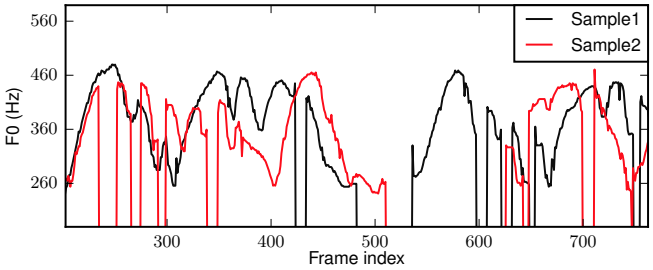
To verify the hypothesis, another DAR was trained after linguistic features related to the Japanese pitch accent were removed. The objective results are listed in the last three rows of Table III, and the sampled F0 contours are shown in Figure 12 (b). Interestingly, Figure 12 (b) shows that sampled contours occasionally deviated from the natural F0 contour, e.g., after the 200th and 500th frames. According to the native Japanese speakers, those F0 curves were perceived as either unnatural segments or different accents from the natural ones. These results indicate that DAR's performance in this experiment benefited from the relatively accurate input linguistic features.



(a) DAR trained with full set of linguistic features



(b) DAR trained without pitch-accent linguistic features



(c) DAR trained without any linguistic feature

Fig. 12. Randomly sampled F0 contours from DAR ($P_d = 0.5$)

We believe that these results are somewhat consistent with those from our previous experiment on an English corpus, in which the proposed DAR sampled more distinguishable F0 contours every time. In English TTS systems, what we can obtain from a lexicon is the lexical stress. However, it does not sufficiently explain an F0 contour. For specifying the general shape of the F0 contour in English, we need to have accurate English pitch-accent information[2], but the English pitch accents cannot be perfectly inferred from the text [85]. Therefore, it is thought that the degrees of informativeness of

[2]We follow the literature to use the term 'pitch accent' for both English and Japanese. However, there are fundamental differences between 'pitch accent' in English and Japanese [84]

the English input linguistic features allowed the English DAR to generate varied F0 contours through sampling.

Finally, it is interesting to see what would happen if no linguistic feature is provided for DAR. This was implemented by setting the input sequence $\boldsymbol{x}_{1:T}$ to zero during both training and generation. Two samples randomly drawn from such a DAR are shown in Figure 12 (c). Although these two F0 contours were nonsense, they were smooth and resembled the high-low movement of natural F0 contours. As RMDN and SAR could not generate smooth random samples under the same condition, this result provides further evidence of DAR's ability to model the temporal correlation of F0 contours.

VI. Conclusion

We investigated the task of F0 modeling for TTS from the perspective of sequential data modeling. On the basis of the analysis and experiments, it was demonstrated that a conventional RNN may not properly capture the temporal dependency of the F0 contour across frames. Even the F0s of two adjacent frames are assumed to be statistically independent. As a result, the RNN generated very noisy F0 contours when a sampling-based generation method was used. Although the RNN could generate smooth F0 contours by using a conventional mean-based generation method, the generated F0 contours became over-smoothed and made the synthetic speech tedious in perception.

We attempted to improve an RNN by adding the AR dependency to the model. The first attempt was based on our recently proposed SAR. However, although this SAR takes into account the AR dependency within a local time window, it was shown that the SAR also generated noisy F0 contours when the sampling-based generation method was used. The weakness of the SAR is due to the fact that it only relies on linear transformation to capture the AR dependency.

Therefore, in this paper, we further proposed a DAR for accurate F0 modeling. The basic idea of this model is to feed back the previous F0 observation as the input to a recurrent layer. This feedback link can propagate the previous F0 data to all the following frames through non-linear transformation. To make the model practical, we also proposed quantized F0 representation and a data dropout strategy of the feedback link. As experiments demonstrated, the proposed DAR generated smooth and quite natural F0 contours even if random sampling was used. This result has not been achieved with other F0 models. Furthermore, when the conventional mean-based generation method was used, the DAR generated F0 contours with an appropriate dynamic range and high accuracy. It also outperformed the SAR, an RMDN, and an RNN in a subjective evaluation test on the prosodic naturalness of synthetic speech.

References

[1] P. Taylor, *Text-to-Speech Synthesis*. Cambridge University Press, 2009.

[2] 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, 2013.

[3] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, pp. 1039–1064, 2009.

[4] C. Gussenhoven, *The phonology of tone and intonation*. Cambridge University Press, 2004.

[5] M. E. Beckman and G. Ayers, "Guidelines for ToBI labelling," *The OSU Research Foundation*, vol. 3, 1997. [Online]. Available: http://www.ling.ohio-state.edu//research/phonetics/E_ToBI/singer_tobi.html

[6] Z. H. Ling, S. Y. Kang, H. Zen, A. Senior, M. Schuster, X. J. Qian, H. M. Meng, and L. Deng, "Deep learning for acoustic modeling in parametric speech generation: A systematic review of existing techniques and future trends," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 35–52, 2015.

[7] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. ICASSP*, 2013, pp. 7962–7966.

[8] Y. Fan, Y. Qian, F. Xie, and F. K. Soong, "TTS synthesis with bidirectional LSTM based recurrent neural networks," in *Proc. Interspeech*, 2014, pp. 1964–1968.

[9] 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, 2011.

[10] J. Pierrehumbert, "Synthesizing intonation," *The Journal of the Acoustical Society of America*, vol. 70, no. 4, pp. 985–995, 1981.

[11] J. R. Bellegarda, K. E. Silverman, K. Lenzo, and V. Anderson, "Statistical prosodic modeling: from corpus design to parameter estimation," *IEEE Trans. on speech and audio processing*, vol. 9, no. 1, pp. 52–66, 2001.

[12] A. Black and A. Hunt, "Generating F0 contours from ToBI labels using linear regression," in *Proc. ICSLP*, vol. 3, 1996, pp. 1385–1388.

[13] K. N. Ross and M. Ostendorf, "A dynamical system model for generating fundamental frequency for speech synthesis," *IEEE Trans. on speech and audio processing*, vol. 7, no. 3, pp. 295–309, 1999.

[14] Y. Sagisaka, "On the prediction of global F0 shape for Japanese text-to-speech," in *Proc. ICASSP*, 1990, pp. 325–328.

[15] M. S. Scordilis and J. N. Gowdy, "Neural network based generation of fundamental frequency contours," in *Proc. ICASSP*, 1989, pp. 219–222 vol.1.

[16] C. Traber, "F0 generation with a data base of natural F0 patterns and with a neural network," in *Proc. ESCA Workshop on Speech Synthesis*, 1991, pp. 141–144.

[17] S. H. Chen, S. H. Hwang, and Y. R. Wang, "An RNN-based prosodic information synthesizer for Mandarin text-to-speech," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 3, pp. 226–239, 1998.

[18] J. Buhmann, H. Vereecken, J. Fackrell, J.-P. Martens, and B. Van Coile, "Data driven intonation modelling of 6 languages," in *Proc. ICSLP*, 2000, pp. 179–182.

[19] R. Fernandez, A. Rendel, B. Ramabhadran, and R. Hoory, "Prosody contour prediction with long short-term memory, bi-directional, deep recurrent neural networks." in *Proc. Interspeech*, 2014, pp. 2268–2272.

[20] X. Wang, S. Takaki, and J. Yamagishi, "An autoregressive recurrent mixture density network for parametric speech synthesis," in *Proc. ICASSP*, 2017, pp. 4895–4899.

[21] ——, "An RNN-based quantized F0 model with multi-tier feedback links for text-to-speech synthesis," in *Proc. Interspeech*, 2017, pp. 1059–1063.

[22] H. Fujisaki and K. Hirose, "Analysis of voice fundamental frequency contours for declarative sentences of Japanese," *Journal of the Acoustical Society of Japan (E)*, vol. 5, no. 4, pp. 233–242, 1984.

[23] H. Kameoka, J. Le Roux, and Y. Ohishi, "A statistical model of speech F0 contours," in *Proc. Interspeech*, 2010, pp. 43–48.

[24] K. Hirose, K. Sato, Y. Asano, and N. Minematsu, "Synthesis of F0 contours using generation process model parameters predicted from unlabeled corpora: Application to emotional speech synthesis," *Speech communication*, vol. 46, no. 3, pp. 385–404, 2005.

[25] P. Taylor, "Analysis and synthesis of intonation using the tilt model," *JASA*, vol. 107, no. 3, pp. 1697–1714, 2000.

[26] K. E. Dusterhoff, A. W. Black, and P. A. Taylor, "Using decision trees within the Tilt intonation model to predict F0 contours." *Proc. Eurospeech*, pp. 1627–1630, 1999.

[27] S. Prom-On, Y. Xu, and B. Thipakorn, "Modeling tone and intonation in Mandarin and English as a process of target approximation," *JASA*, vol. 125, no. 1, pp. 405–424, 2009.

[28] X. Sun, "F0 generation for speech synthesis using a multi-tier approach," in *Proc. ICSLP*, 2002, pp. 2077–2080.

[29] L. Gao, Z.-H. Ling, L.-H. Chen, and L.-R. Dai, "Improving F0 prediction using bidirectional associative memories and syllable-level F0 features for HMM-based Mandarin speech synthesis," in *Proc. ISCSLP*. IEEE, 2014, pp. 275–279.

[30] J. Teutenberg, C. Watson, and P. Riddle, "Modelling and synthesising F0 contours with the discrete cosine transform," in *Proc. ICASSP*, 2008, pp. 3973–3976.

[31] J. Latorre and M. Akamine, "Multilevel parametric-base F0 model for speech synthesis." in *Proc. Interspeech*, 2008, pp. 2274–2277.

[32] N. Obin, A. Lacheret, and X. Rodet, "Stylization and trajectory modelling of short and long term speech prosody variations," in *Proc. Interspeech*, 2011, pp. 2029–2032.

[33] Y. Qian, Z. Wu, B. Gao, and F. K. Soong, "Improved prosody generation by maximizing joint probability of state and longer units," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 19, no. 6, pp. 1702–1710, 2011.

[34] X. Yin, M. Lei, Y. Qian, F. K. Soong, L. He, Z.-H. Ling, and L.-R. Dai, "Modeling DCT parameterized F0 trajectory at intonation phrase level with DNN or decision tree," in *Proc. Interspeech*, 2014, pp. 2273–2277.

[35] M. S. Ribeiro and R. A. J. Clark, "A multi-level representation of F0 using the continuous wavelet transform and the discrete cosine transform," in *Proc. ICASSP*, 2015, pp. 4909–4913.

[36] Y. Asano, M. Gubian, and D. Sacha, "Cutting down on manual pitch contour annotation using data modelling," in *Proc. Speech Prosody*, 2016, pp. 282–286.

[37] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi, "Multi-space probability distribution HMM," *IEICE Trans. on Information and Systems*, vol. 85, no. 3, pp. 455–464, 2002.

[38] H. Zen and N. Braunschweiler, "Context-dependent additive log F0 model for hmm-based speech synthesis," in *Proc. Interspeech*, 2009, pp. 2091–2094.

[39] M. Lei, Y. Wu, F. K. Soong, Z. H. Ling, and L. Dai, "A hierarchical F0 modeling method for HMM-based speech synthesis," in *Proc. Interspeech*, 2010, pp. 2170–2173.

[40] X. Wang, S. Takaki, and J. Yamagishi, "Investigating very deep highway networks for parametric speech synthesis," *Speech Communication*, vol. 96, pp. 1 – 9, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167639316303703

[41] C. M. Bishop, *Neural networks for pattern recognition*. Oxford university press, 1995.

[42] ——, "Mixture Density Networks," Aston University, Tech. Rep., 2004. [Online]. Available: http://eprints.aston.ac.uk/373/

[43] M. Schuster, "Better generative models for sequential data problems: Bidirectional recurrent mixture density networks," in *Proc. NIPS*, 1999, pp. 589–595.

[44] B. C. J. Moore, *An Introduction to the Psychology of Hearing*, 6th ed. Brill, 2012.

[45] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[46] W. Ling, C. Dyer, A. W. Black, I. Trancoso, R. Fermandez, S. Amir, L. Marujo, and T. Luis, "Finding function in form: Compositional character models for open vocabulary word representation," in *Proc. EMNLP*, 2015, pp. 1520–1530.

[47] J. Chiu and E. Nichols, "Named entity recognition with bidirectional lstm-cnns," *Transactions of the Association of Computational Linguistics*, vol. 4, no. 1, pp. 357–370, 2016.

[48] H. Zen, M. J. Gales, Y. Nankaku, and K. Tokuda, "Product of experts for statistical parametric speech synthesis," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 20, no. 3, pp. 794–805, 2012.

[49] H. Zen, K. Tokuda, and T. Kitamura, "Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic feature vector sequences," *Computer Speech & Language*, vol. 21, no. 1, pp. 153–173, 2007.

[50] K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Trajectory training considering global variance for speech synthesis based on neural networks," in *Proc. ICASSP*, 2016, pp. 5600–5604.

[51] B. Frey, *Graphical Models for Machine Learning and Digital Communication*, ser. A Bradford book. Bradford book, 1998.

[52] J. Makhoul, "Linear prediction: A tutorial review," *Proceedings of the IEEE*, vol. 63, no. 4, pp. 561–580, 1975.

[53] M. Shannon, H. Zen, and W. Byrne, "Autoregressive models for statistical parametric speech synthesis," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 21, no. 3, pp. 587–597, 2013.

[54] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," in *Proc. NIPS*, 2015, pp. 2980–2988.

[55] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model." in *Proc. Interspeech*, vol. 2, 2010, p. 3.

[56] K. Gregor, I. Danihelka, A. Mnih, C. Blundell, and D. Wierstra, "Deep autoregressive networks," in *Proc. ICML*, 2014, pp. 1242–1250.

[57] H. Larochelle and I. Murray, "The neural autoregressive distribution estimator," in *Proc. AISTATS*, 2011, pp. 29–37.

[58] B. Uria, M.-A. Côté, K. Gregor, I. Murray, and H. Larochelle, "Neural autoregressive distribution estimation," *Journal of Machine Learning Research*, vol. 17, no. 205, pp. 1–37, 2016.

[59] A. Van Den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *Proc. ICML*, 2016, pp. 1747–1756.

[60] B. Uria, I. Murray, S. Renals, C. Valentini-Botinhao, and J. Bridle, "Modelling acoustic feature dependencies with artificial neural networks: Trajectory-RNADE," in *Proc. ICASSP*, 2015, pp. 4465–4469.

[61] M. Shannon, H. Zen, and W. Byrne, "The effect of using normalized models in statistical speech synthesis," in *Proc. Interspeech*, 2011.

[62] H. Zen and H. Sak, "Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis," in *Proc. ICASSP*, 2015, pp. 4470–4474.

[63] F. Morin and Y. Bengio, "Hierarchical probabilistic neural network language model," in *Proc. AISTATS*, vol. 5, 2005, pp. 246–252.

[64] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural computation*, vol. 1, no. 2, pp. 270–280, 1989.

[65] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, "Sequence level training with recurrent neural networks," *Proc. ICLR*, 2016. [Online]. Available: https://arxiv.org/pdf/1511.06732.pdf

[66] X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel, "Variational lossy autoencoder," in *Proc. ICLR*, 2017, pp. –.

[67] A. Venkatraman, M. Hebert, and J. A. Bagnell, "Improving multi-step prediction of learned time series models." in *Proc. AAAI*, 2015, pp. 3024–3030.

[68] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Proc. NIPS*, 2015, pp. 1171–1179.

[69] F. Huszár, "How (not) to train your generative model: Scheduled sampling, likelihood, adversary?" *arXiv preprint arXiv:1511.05101*, 2015.

[70] S. Shen, Y. Cheng, Z. He, W. He, H. Wu, M. Sun, and Y. Liu, "Minimum risk training for neural machine translation," in *Proc. ACL (Volume 1: Long Papers)*, vol. 1, 2016, pp. 1683–1692.

[71] H. Kawai, T. Toda, J. Ni, M. Tsuzaki, and K. Tokuda, "XIMERA: A new TTS from ATR based on corpus-based technologies," in *Proc. SSW5*, 2004, pp. 179–184.

[72] X. Wang, S. Takaki, and J. Yamagishi, "A comparative study of the performance of HMM, DNN, and RNN based speech synthesis systems trained on very large speaker-dependent corpora," in *Proc. SSW9*, 2016, pp. 125–128.

[73] HTS Working Group, "The Japanese TTS System 'Open JTalk'," 2015. [Online]. Available: http://open-jtalk.sourceforge.net/

[74] T. Kudo, "MeCab: Yet Another Part-of-Speech and Morphological Analyzer." [Online]. Available: http://mecab.sourceforge.net/

[75] HTS Working Group, "An example of context-dependent label format for HMM-based speech synthesis in Japanese," 2015. [Online]. Available: http://hts.sp.nitech.ac.jp/archives/2.3/HTS-demo_NIT-ATR503-M001.tar.bz2

[76] L. Juvela, X. Wang, S. Takaki, S. Kim, M. Airaksinen, and J. Yamagishi, "The NII speech synthesis entry for Blizzard Challenge 2016," in *Proc. Blizzard Challenge Workshop*, 2016.

[77] D. O'Shaughnessy, *Speech Communications: Human and Machine*. Institute of Electrical and Electronics Engineers, 2000.

[78] K. Tokuda, T. Kobayashi, T. Masuko, and S. Imai, "Mel-generalized cepstral analysis a unified approach," in *Proc. ICSLP*, 1994, pp. 1043–1046.

[79] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: A vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Trans. on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[80] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[81] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. AISTATS*, 2010, pp. 249–256.

[82] F. Weninger, J. Bergmann, and B. Schuller, "Introducing CURRENT: The Munich open-source CUDA recurrent neural network toolkit," *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 547–551, 2015.

[83] N. Minematsu, R. Kuroiwa, K. Hirose, and M. Watanabe, "CRF-based statistical learning of Japanese accent sandhi for developing Japanese text-to-speech synthesis systems," in *Proc. SSW6*, 2007.

[84] J. J. Venditti, "The J_ToBI model of Japanese intonation," *Prosodic typology: The phonology of intonation and phrasing*, pp. 172–200, 2005.

[85] J. Hirschberg, "Pitch accent in context predicting intonational prominence from text," *Artificial Intelligence*, vol. 63, no. 1, pp. 305–340, 1993.

PLACE PHOTO HERE

Xin Wang From October 2015, he is a PhD student at National Institute of Informatics. He received the M.S. degree from University of Science and Technology of China in 2015 because of his work on HMM-based speech synthesis. His research interests include statistical speech synthesis and machine learning.

PLACE PHOTO HERE

Shinji Takaki receieved the B.E. degree in computer science, the M.E. and Ph.D. degrees in scientific and engineering simulation from Nagoya Institute of Technology, Nagoya, Japan in 2009, 2011, and 2014, respectively. From September 2013 to January 2014, he was a visiting researcher at University of Edinburgh University. From April 2014, he is a project researcher at National Institute of Informatics. His research interests include statistical machine learning and speech synthesis. He is a member of the Acoustical Society of Japan and information processing society of Japan.

PLACE PHOTO HERE

Junichi Yamagishi (SM'13) is an associate professor of National Institute of Informatics in Japan. He is also a senior research fellow in the Centre for Speech Technology Research (CSTR) at the University of Edinburgh, UK. He was awarded a Ph.D. by Tokyo Institute of Technology in 2006 for a thesis that pioneered speaker-adaptive speech synthesis and was awarded the Tejima Prize as the best Ph.D. thesis of Tokyo Institute of Technology in 2007. Since 2006, he has authored and co-authored over 100 refereed papers in international journals and conferences. He was awarded the Itakura Prize from the Acoustic Society of Japan, the Kiyasu Special Industrial Achievement Award from the Information Processing Society of Japan, and the Young Scientists' Prize from the Minister of Education, Science and Technology, the JSPS prize in 2010, 2013, 2014, and 2016, respectively.

He was one of organizers for special sessions on "Spoofing and Countermeasures for Automatic Speaker Verification" at Interspeech 2013, "ASVspoof evaluation" at Interspeech 2015, "Voice conversion challenge 2016" at Interspeech 2016, and "2nd ASVspoof evaluation" at Interspeech 2017. He has been a member of the Speech & Language Technical Committee (SLTC). He served as an Associate Editor of the IEEE/ACM Transactions on Audio, Speech and Language Processing and a Lead Guest Editor for the IEEE Journal of Selected Topics in Signal Processing (JSTSP) special issue on Spoofing and Countermeasures for Automatic Speaker Verification.