DIRECTLY MODELING VOICED AND UNVOICED COMPONENTS IN SPEECH WAVEFORMS BY NEURAL NETWORKS

Keiichi Tokuda[†‡] *Heiga Zen*[†]

[†] Google [‡] Nagoya Institute of Technology, Japan

ABSTRACT

This paper proposes a novel acoustic model based on neural networks for statistical parametric speech synthesis. The neural network outputs parameters of a non-zero mean Gaussian process, which defines a probability density function of a speech waveform given linguistic features. The mean and covariance functions of the Gaussian process represent deterministic (voiced) and stochastic (unvoiced) components of a speech waveform, whereas the previous approach considered the unvoiced component only. Experimental results show that the proposed approach can generate speech waveforms approximating natural speech waveforms.

***Index Terms*—** Statistical parametric speech synthesis; neural network; wavefom

1. INTRODUCTION

Typical statistical parametric speech synthesis systems first extract a set of parametric representation of speech (*e.g.*, cepstra [1], line spectrum pairs [2], fundamental frequency, and aperiodicity [3]) then model relationships between the extracted acoustic parameters and linguistic features associated with the speech waveform using an acoustic model [6] (*e.g.*, hidden Markov models [4], neural networks [5]). There have been a couple of attempts to integrate acoustic feature extraction into acoustic modeling, such as the log spectral distortion-version of minimum generation error training [7], statistical vocoder [8], waveform-level statistical model [9], and mel-cepstral analysis-integrated hidden Markov models [10].

Tokuda and Zen recently proposed a neural network-based approach to integrate acoustic feature extraction into acoustic modeling [11]. Here, a neural network outputs parameters of a *zero* mean Gaussian process, which defines a probability density function of a speech waveform given linguistic features. The covariance function of the Gaussian process is parameterized by minimum-phase cepstrum. The network weights are optimized so as to maximize the log likelihood of the Gaussian process given corresponding pairs of speech waveforms (target) and linguistic feature sequences (input). This approach can overcome the limitations of the previous approaches, such as two-step optimization (acoustic feature extraction → acoustic modeling), use of spectra rather than waveforms, and use of overlapping and shifting frames as unit. However, the speech signal model used in this approach has only a stochastic (unvoiced) component, whereas human speech has both stochastic and deterministic (voiced) components.

This paper extends the previous approach [11] to have both the voiced and unvoiced components in its speech signal model. A neural network outputs parameters of a *non-zero* mean Gaussian process, which defines a probability density function of a speech waveform given linguistic features. The deterministic (voiced) component of a speech waveform is modeled by the mean function of the Gaussian process, which is parameterized by mixed-phase complex cepstrum. Its training algorithm which can run sequentially in a sample-by-sample or segment-by-segment manner is also derived.

The rest of the paper is organized as follows. Section 2 defines the signal model and the waveform-level probability density function. Section 3 derives the training algorithm. Preliminary experimental results are presented in Section 4. Concluding remarks are given in the final section.

2. WAVEFORM-LEVEL DEFINITION OF PROBABILITY DENSITY FUNCTION OF SPEECH

2.1. Signal model

This paper adopts the signal model shown in Fig. 1. Thus, the probability density function of a discrete-time speech signal $\boldsymbol{x} = [x(0), x(1), \ldots, x(T-1)]^\top$ corresponding to an utterance or whole speech database is assumed to be a *non-zero mean* stationary Gaussian process, which can be written as

$$p(\boldsymbol{x} \mid \lambda) = \mathcal{N}\left(\boldsymbol{x}; \boldsymbol{H}_{\boldsymbol{c}_v}\boldsymbol{p}, \boldsymbol{\Sigma}_{\boldsymbol{c}_u}\right), \tag{1}$$

where λ is the model parameter set, $\boldsymbol{p} = [p(0), p(1), \ldots, p(T-1)]^\top$ is a pulse sequence having value 1 at pitch mark positions otherwise 0:

$$\boldsymbol{p} = [0, \ldots, 0, 1, 0, \ldots, 0, 1, 0, \ldots, 0, 1, 0, \ldots, 0]^\top, \tag{2}$$

and $\boldsymbol{H}_{\boldsymbol{c}_v}$ is a deterministic component matrix[1] given as

$$\boldsymbol{H}_{\boldsymbol{c}_v} = \begin{bmatrix} h(0) & h(-1) & \cdots & h(-T+1) \\ h(1) & h(0) & \ddots & \vdots \\ \vdots & \ddots & \ddots & h(-1) \\ h(T-1) & \cdots & h(1) & h(0) \end{bmatrix}, \tag{3}$$

whose elements are given by the impulse response of the mixed phase system function $H_v(z)$:

$$h(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_v(e^{j\omega})\, e^{j\omega n}\, d\omega. \tag{4}$$

In this paper, we assume that the system function $H_v(z)$ generating voiced component $v(t)$ is parameterized by complex cepstrum $\boldsymbol{c}_v$ as

$$H_v(e^{j\omega}) = \exp \sum_{m=-M}^{M} c_v(m)\, e^{-j\omega m}, \tag{5}$$

where $\boldsymbol{c}_v = [c_v(-M), ..., c_v(0), \ldots, c_v(M)]^\top$. The system $H_v(z)$ should not model *delay* since it causes an under-determined problem

[1] Although we assume that $\boldsymbol{x}$ and $\boldsymbol{p}$ are infinite sequences, they are described as finite sequences for notation simplicity. When they are finite sequences, $\boldsymbol{H}$ should be a circulant matrix rather than a Toeplitz matrix.

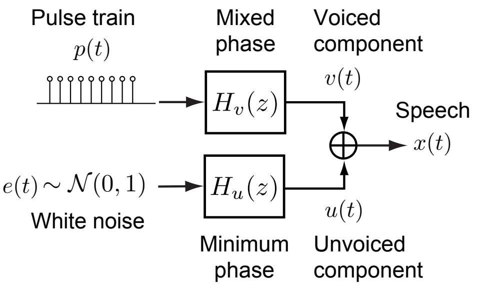

Fig. 1. Speech signal model.

when we estimate $H_v(z)$ and pulse positions of $p(t)$ simultaneously. The complex cepstral representation can avoid the problem because it intrinsically does not represent *delay* of the system.

The covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{c}_u}$ is given as

$$\boldsymbol{\Sigma}_{\boldsymbol{c}_u} = \begin{bmatrix} r(0) & r(1) & \cdots & r(T-1) \\ r(1) & r(0) & \ddots & \vdots \\ \vdots & \ddots & \ddots & r(1) \\ r(T-1) & \cdots & r(1) & r(0) \end{bmatrix} \tag{6}$$

where

$$r(k) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| H_u(e^{j\omega}) \right|^2 e^{j\omega k} \, d\omega, \tag{7}$$

and $\left| H_u(e^{j\omega}) \right|^2$ is the power spectrum of the unvoiced component $u(t)$. This paper assumes that the corresponding minimum-phase system function $H_u(z)$ is parameterized by minimum cepstrum $\boldsymbol{c}_u$ as

$$H_u(e^{j\omega}) = \exp \sum_{m=0}^{M} c_u(m) \, e^{-j\omega m}, \tag{8}$$

where $\boldsymbol{c}_u = [c_u(0), c_u(1), c_u(2), \ldots, c_u(M)]^\top$. The inverse of the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{c}}$ can be written as the same form in [11] as

$$\boldsymbol{\Sigma}_{\boldsymbol{c}_u}^{-1} = \boldsymbol{A}_{\boldsymbol{c}_u}^\top \boldsymbol{A}_{\boldsymbol{c}_u}, \tag{9}$$

where

$$\boldsymbol{A}_{\boldsymbol{c}_u} = \begin{bmatrix} a(0) & 0 & \cdots & 0 \\ a(1) & a(0) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ a(T-1) & \cdots & a(1) & a(0) \end{bmatrix} \tag{10}$$

and $a(n)$ is the impulse response of the inverse system given as

$$a(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_u^{-1}(e^{j\omega}) \, e^{j\omega n} \, d\omega. \tag{11}$$

From the above definition, the logarithm of the probability density function can be written as

$$\begin{aligned} \log p(\boldsymbol{x} \mid \lambda) = & -\frac{T}{2} \log 2\pi + \frac{1}{2} \log \left| \boldsymbol{A}_{\boldsymbol{c}_u}^\top \boldsymbol{A}_{\boldsymbol{c}_u} \right| \\ & - \frac{1}{2} \left(\boldsymbol{x} - \boldsymbol{H}_{\boldsymbol{c}_v} \boldsymbol{p} \right)^\top \boldsymbol{A}_{\boldsymbol{c}_u}^\top \boldsymbol{A}_{\boldsymbol{c}_u} \left(\boldsymbol{x} - \boldsymbol{H}_{\boldsymbol{c}_v} \boldsymbol{p} \right) \end{aligned} \tag{12}$$

where the model parameter set is given as $\lambda = \{\boldsymbol{c}_v, \boldsymbol{c}_u, \boldsymbol{p}\}$. We assume that pulse positions in $\boldsymbol{p}$ are extracted by using an external pitch marker and therefore $\boldsymbol{p}$ is fixed in the following discussion.

2.2. Non-stationarity modeling

Equation (12) can be rewritten as

$$\begin{aligned} \log p(\boldsymbol{x} \mid \lambda) = & -\frac{T}{2} \log 2\pi + \frac{1}{2} \log \left| \boldsymbol{A}_{\boldsymbol{c}_u}^\top \boldsymbol{A}_{\boldsymbol{c}_u} \right| \\ & - \frac{1}{2} \left(\boldsymbol{A}_{\boldsymbol{c}_u} \boldsymbol{x} - \boldsymbol{G} \boldsymbol{p} \right)^\top \left(\boldsymbol{A}_{\boldsymbol{c}_u} \boldsymbol{x} - \boldsymbol{G} \boldsymbol{p} \right), \end{aligned} \tag{13}$$

where $\boldsymbol{G} = \boldsymbol{A}_{\boldsymbol{c}_u} \boldsymbol{H}_{\boldsymbol{c}_v}$ is given as

$$\boldsymbol{G} = \begin{bmatrix} g(0) & g(-1) & \cdots & g(-T+1) \\ g(1) & g(0) & \ddots & \vdots \\ \vdots & \ddots & \ddots & g(-1) \\ g(T-1) & \cdots & g(1) & g(0) \end{bmatrix} \tag{14}$$

and $g(n)$ is the impulse response of the system function $G(z) = H_v(z) H_u^{-1}(z)$:

$$G(e^{j\omega}) = \exp \sum_{m=-M}^{M} \left\{ c_v(m) - c_u(m) \right\} e^{-j\omega m}, \tag{15}$$

$$(c_u(m) = 0, m < 0),$$

that is,

$$g(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} G(e^{j\omega}) \, e^{j\omega n} \, d\omega. \tag{16}$$

To model the non-stationary nature of the speech signal, $\boldsymbol{x}$ is assumed to be segment-by-segment piecewise-stationary: $\boldsymbol{A}_{\boldsymbol{c}_u}$ in Eq. (10) and $\boldsymbol{G}$ in Eq. (14) are redefined as

$$\boldsymbol{A}_{\boldsymbol{c}_u} = \left[\begin{array}{cccccccc} \ddots & \ddots & & & & & & \\ & a^{(i-1)}(0) & 0 & \cdots & \cdots & \cdots & \cdots & \cdots \\ \hline \cdots & a^{(i)}(1) & a^{(i)}(0) & 0 & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & a^{(i)}(1) & a^{(i)}(0) & 0 & \cdots & \cdots & \cdots \\ & & & \ddots & \ddots & \ddots & & \\ \cdots & \cdots & \cdots & \cdots & a^{(i)}(1) & a^{(i)}(0) & 0 & \cdots \\ \hline \cdots & \cdots & \cdots & \cdots & \cdots & a^{(i+1)}(1) & a^{(i+1)}(0) & \\ & & & & & & \ddots & \ddots \end{array} \right] \Bigg\} L \tag{17}$$

and

$$\boldsymbol{G} = \left[\begin{array}{cccccccc} \ddots & \ddots & & & & & & \\ & g^{(i-1)}(0) & g^{(i-1)}(-1) & \cdots & \cdots & \cdots & \cdots & \cdots \\ \hline \cdots & g^{(i)}(1) & g^{(i)}(0) & g^{(i)}(-1) & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & g^{(i)}(1) & g^{(i)}(0) & g^{(i)}(-1) & \cdots & \cdots & \cdots \\ & & & \ddots & \ddots & \ddots & & \\ \cdots & \cdots & \cdots & \cdots & g^{(i)}(1) & g^{(i)}(0) & g^{(i)}(-1) & \cdots \\ \hline \cdots & \cdots & \cdots & \cdots & \cdots & g^{(i+1)}(1) & g^{(i+1)}(0) & \\ & & & & & & \ddots & \ddots \end{array} \right] \Bigg\} L, \tag{18}$$

where i is the segment index, L is the size of each segment, $a^{(i)}(n)$ is the impulse response of the inverse system of $H_u^{(i)}(z)$ represented by cepstrum

$$\boldsymbol{c}_u^{(i)} = \left[c_u^{(i)}(0), c_u^{(i)}(1), \ldots, c_u^{(i)}(M) \right]^\top, \tag{19}$$

as in Eq. (8) for the i-th segment, and $g^{(i)}(n)$ is the impulse response of the system $G^{(i)}(z)$ represented by cepstrum $\boldsymbol{c}_u^{(i)}$ and

$$\boldsymbol{c}_v^{(i)} = \left[c_v^{(i)}(-M), \ldots, c_v^{(i)}(0), \ldots, c_v^{(i)}(M) \right]^\top \tag{20}$$

as in Eq. (15) for the i-th segment.

Here the model parameter set of the probability density function $p(\boldsymbol{x} \mid \lambda)$ can be written as $\lambda = \boldsymbol{c} = \{\boldsymbol{c}_v, \boldsymbol{c}_u\}$, where $\boldsymbol{c}_v = \left\{\boldsymbol{c}_v^{(0)}, \boldsymbol{c}_v^{(1)}, \ldots, \boldsymbol{c}_v^{(I-1)}\right\}, \boldsymbol{c}_u = \left\{\boldsymbol{c}_u^{(0)}, \boldsymbol{c}_u^{(1)}, \ldots, \boldsymbol{c}_u^{(I-1)}\right\}$, and I is the number of segments in $\boldsymbol{x}$ corresponding to an utterance or whole speech database, and thus $T = L \times I$. Note that $\boldsymbol{p}$ is omitted from λ since it is assumed to be fixed in this paper.

3. TRAINING ALGORITHM

3.1. Derivative of the log likelihood

With some elaboration,[2] the partial derivative of Eq. (13) w.r.t. $\boldsymbol{c}_v^{(i)}$ can be derived as $\boldsymbol{d}_v^{(i)} = \partial \log p(\boldsymbol{x} \mid \boldsymbol{c}) / \partial \boldsymbol{c}_v^{(i)} = \left[d_v^{(i)}(-M), \ldots, d_v^{(i)}(0), \ldots, d_v^{(i)}(M)\right]^\top$, where

$$d_v^{(i)}(m) = \sum_{k=0}^{L-1} e^{(i)}(Li+k)\, f^{(i)}(Li+k-m), \tag{21}$$

and $f^{(i)}(t)$ is the output of $G^{(i)}(z)$, whose input is $p(t)$, *i.e.*

$$f^{(i)}(t) = \sum_{n=\infty}^{\infty} g^{(i)}(n)\, p(t-n), \tag{22}$$

The signal $e^{(i)}(t)$ is given as

$$e^{(i)}(t) = s^{(i)}(t) - f^{(i)}(t), \tag{23}$$

where $s^{(i)}(t)$ is the output of the inverse of $H_u^{(i)}(z)$, whose input is $x(t)$, *i.e.*

$$s^{(i)}(t) = \sum_{n=0}^{\infty} a^{(i)}(n)\, x(t-n). \tag{24}$$

The partial derivative of Eq. (13) w.r.t. $\boldsymbol{c}_u^{(i)}$ can also be derived as $\boldsymbol{d}_u^{(i)} = \partial \log p(\boldsymbol{x} \mid \boldsymbol{c}) / \partial \boldsymbol{c}_u^{(i)} = \left[d_u^{(i)}(0), d_u^{(i)}(1), \ldots, d_u^{(i)}(M)\right]^\top$, where

$$d_u^{(i)}(m) = \sum_{k=0}^{L-1} e^{(i)}(Li+k)\, e^{(i)}(Li+k-m) - \delta(m)L, \tag{25}$$

and $\delta(m)$ is the unit impulse function.

3.2. Sequential algorithm

For calculating the impulse response $a^{(i)}(n)$ or $g^{(i)}(n)$ using a recursive formula [13], $\mathcal{O}(MN)$ operations are required at each segment i, even if it is truncated with a sufficiently large number of N. Furthermore, for calculating $s^{(i)}(t)$ in Eq. (24) or $f^{(i)}(t)$ in Eq. (22), $\mathcal{O}(N(M+L))$ operations are required for each segment i.

First, to reduce the computational burden in calculating $s^{(i)}(t)$ in Eq. (24), the following two approximations are applied;

1. By assuming

$$s^{(i)}(t) \simeq e^{(i-1)}(t), \quad t = Li-M, \ldots, Li-1 \tag{26}$$

$s^{(i)}(t)$ can be calculated as the output of the inverse system whose parameters change segment by segment as follows:

$$s^{(i)}(t) = s(t) = \sum_{n=0}^{\infty} a_t(n)\, x(t-n), \tag{27}$$

where

$$a_t(n) = a^{(i)}(n), \quad t = Li, \ldots, Li+L-1 \tag{28}$$

2. As an approximation, inverse filtering in Eq. (27) can be efficiently calculated by the log magnitude approximation (LMA) filter[3] [12] whose coefficients are given by

$$-\boldsymbol{c}_{u\,t} = -\boldsymbol{c}_u^{(i)}, \quad t = Li, \ldots, Li+L-1 \tag{29}$$

The same approximation can be applied to calculation of $s^{(i)}(t)$ in Eq. (24), except that the system function $G(z)$ corresponding to Eq. (22) is decomposed into minimum- and maximum-phase components as $G(z) = G_+(z)G_-(z)$, where

$$G_+(e^{j\omega}) = \exp \sum_{m=0}^{M} \{c_v(m) - c_u(m)\}\, e^{-j\omega m}, \tag{30}$$

$$G_-(e^{j\omega}) = \exp \sum_{m=-M}^{-1} c_v(m)\, e^{-j\omega m}. \tag{31}$$

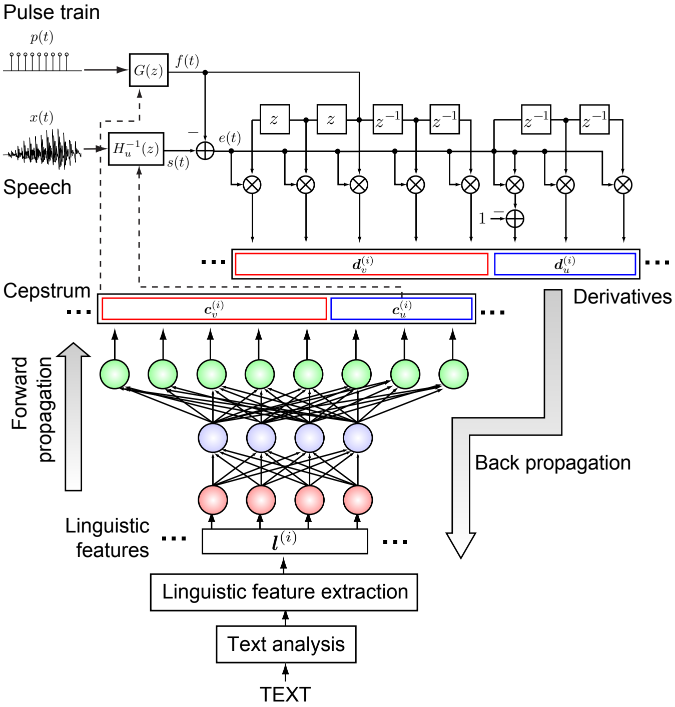

Fig. 2. Block diagram of the proposed waveform-based framework ($M = 2$, $L = 1$, *i.e.* $i = t$). The element z can be realized in the training phase because it is in an *offline* processing mode. For notation simplicity, here acoustic model is illustrated as a feed-forward neural network rather than long short-term memory recurrent neural network (LSTM-RNN).

[2] Similar derivation can be found in Eqs. (14) and (16) in [12].

[3] The LMA filter is a special type of digital filter which can approximate the system function of Eq. (8).

Each of them are implemented in the LMA filter structure. However, $G_-(z)$ is an anticausal system while $G_+(z)$ is a causal system, and thus $G_-(z)$ should run in a time-reversal manner.

With the above approximations, a simple structure for training the neural network acoustic model, which represents a mapping from linguistic features to speech signals, can be derived. It can run in a sequential manner as shown in Fig. 2. This neural network outputs cepstrum $\boldsymbol{c}$ given linguistic feature vector sequence[4] $\boldsymbol{l} = \left\{ \boldsymbol{l}^{(0)}, \boldsymbol{l}^{(1)}, \ldots, \boldsymbol{l}^{(I-1)} \right\}$, which in turn gives a probability density function of speech signals $\boldsymbol{x}$ corresponding to an utterance or whole speech database conditioned on $\boldsymbol{l}$ as

$$p(\boldsymbol{x} \mid \boldsymbol{l}, \mathcal{M}) = \mathcal{N}\left(\boldsymbol{x}; \boldsymbol{H}_{\boldsymbol{c}(\boldsymbol{l})}\boldsymbol{p}, \boldsymbol{\Sigma}_{\boldsymbol{c}(\boldsymbol{l})}\right), \tag{32}$$

where $\mathcal{M}$ denotes a set of network weights, $\boldsymbol{c}(\boldsymbol{l})$ is given by activations at the output layer of the network given input linguistic features, and the RHS is given by Eq. (12). By back-propagating the derivative of the log likelihood function through the network, the network weights can be updated to maximize the log likelihood.

It should be noted that the proposed approach optimizes the acoustic feature extraction part and acoustic modeling part simultaneously. As a result, better modeling accuracy can be expected.

3.3. Synthesis structure

The speech waveform can be generated by sampling $\boldsymbol{x}$ from the probability density function $p(\boldsymbol{x} \mid \boldsymbol{l}, \mathcal{M})$. It can be done by using the signal model structure shown in Fig. 1. By decomposing $H_v(z)$ into minimum- and maximum-phase components as $H_v(z) = H_{v+}(z)H_{v-}(z)$, the system functions $H_{v+}(z)$, $H_{v-}(z)$ and $H_u(z)$ can be implemented by using the LMA filter structure, where $H_{v+}(z)$ runs in a time-reversal manner. It should be noted that we need an external F_0 predictor for generating the pulse train $p(t)$.

4. EXPERIMENTS

Speech data in US English from a female professional speaker was used for the experiments. The training and development data sets consisted of 35,497 and 100 utterances, respectively. A speaker-dependent unidirectional LSTM-RNN [14] was trained.

The linguistic features derived from speech data, transcriptions, and alignments, included 560 linguistic contexts, 10 numerical features for coarse-coded position of the current frame within the current phoneme, and one numerical feature for duration of the current phoneme.

The speech data was downsampled from 48 kHz to 16 kHz. Then 0–39 mel-cepstrum, 5 band aperiodicity, 1 $\log F_0$, and 1 voiced/unvoiced binary flag were extracted at each frame. Glottal closure instants (GCI) locations were also extracted using REAPER [15]. Both the input and output features were normalized to have zero-mean unit-variance. The architecture of the LSTM-RNN had 1 feed-forward hidden layer with 256 units and rectified linear (ReLU) activation [16] followed by 3 forward-directed LSTMP [17] hidden layers with 512 memory blocks and 256 projection units, 1 feed-forward hidden layer with 256 units and ReLU activation, and an output layer with 47 units[5] and linear activation.

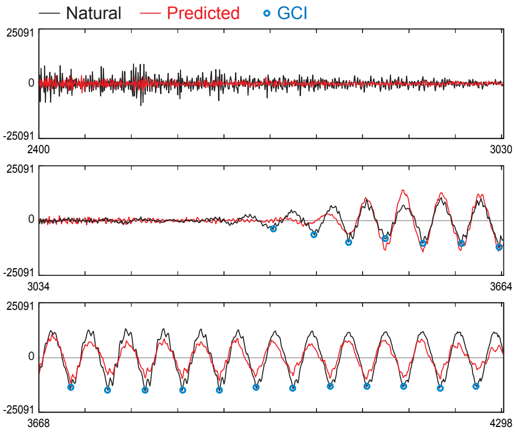

Fig. 3. A segment of the generated speech waveform for a sentence “Two elect only two” not included in the training data.

To reduce the training time and the impact of having many silences, 80% of silence regions were removed. After setting the network weights randomly, they were first updated to minimize the mean squared error between the extracted and predicted acoustic features. Then the last layer was replaced by a randomly initialized output layer with 119 units[6] and linear activation. After updating the weights associated with the output layer, all weights in the network were updated by the proposed sequential algorithm so as to maximize the log likelihood of Eq. (12). They were first updated by non-distributed Adam [18] then distributed AdaGrad [19]. The mini-batch back propagation through time (BPTT) [20] algorithm was used [17] in both cases. Dropout [21] stochastic regularization (50%) was used throughout to prevent overfitting.

Fig. 3 shows a synthesized speech waveform generated from the trained neural network. It can be seen from the figure that a speech waveform approximating the natural speech waveform is generated.

5. CONCLUSIONS

An acoustic modeling approach based on neural networks to statistical parametric speech synthesis was proposed. The network outputs parameters of a *non-zero* mean Gaussian process, which defines a probability density function of a speech waveform given linguistic features. The stochastic (unvoiced) component of a speech waveform is modeled by the covariance function of the Gaussian process, parameterized by minimum-phase cepstrum, whereas the deterministic (voiced) component is modeled by the mean function of the Gaussian process, parameterized by mixed-phase complex cepstrum. Its training algorithm which can run sequentially on speech waveform in a sample-by-sample or segment-by-segment manner was derived.

Future work includes simultaneously estimating pitch marks including fractional pitch search in the model training. One of the limitations of this approach is that both acoustic modeling ($\boldsymbol{l} \rightarrow \boldsymbol{c}$) and waveform modeling ($\boldsymbol{c} \rightarrow \boldsymbol{x}$) error goes to the unvoiced component. Introduction of a covariance structure for the voiced component can alleviate this problem. Performance evaluation in practical conditions as a text-to-speech synthesis application is also included in future work.

[4] The definition of the linguistic feature vector used in this paper can be found in [5] and [14].

[5] It included 0–39 mel-cepstrum, 5 band aperiodicity, 1 $\log F_0$, and 1 voiced/unvoiced flag.

[6] It included 0–39 minimum-phase unvoiced cepstrum, 0–39 minimum-phase voiced cepstrum, and 1–39 maximum-phase voiced cepstrum.

6. REFERENCES

[1] S. Imai and C. Furuichi, “Unbiased estimation of log spectrum,” in *Proc. EURASIP*, 1988, pp. pp.203–206.

[2] F. Itakura, “Line spectrum representation of linear predictor coefficients of speech signals,” *The Journal of the Acoust. Society of America*, vol. 57, no. S1, pp. S35–S35, 1975.

[3] H. Kawahara, J. Estill, and O. Fujimura, “Aperiodicity extraction and control using mixed mode excitation and group delay manipulation for a high quality speech analysis, modification and synthesis system straight,” in *Proc. MAVEBA*, 2001, pp. 13–15.

[4] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, “Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis,” in *Proc. Eurospeech*, 1999, pp. 2347–2350.

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

[6] H. Zen, K. Tokuda, and A. Black, “Statistical parametric speech synthesis,” *Speech Commn.*, vol. 51, no. 11, pp. 1039–1064, 2009.

[7] Y.-J. Wu and K. Tokuda, “Minimum generation error training with direct log spectral distortion on LSPs for HMM-based speech synthesis,” in *Proc. Interspeech*, 2008, pp. 577–580.

[8] T. Toda and K. Tokuda, “Statistical approach to vocal tract transfer function estimation based on factor analyzed trajectory hmm,” in *Proc. ICASSP*, 2008, pp. 3925–3928.

[9] R. Maia, H. Zen, and M. Gales, “Statistical parametric speech synthesis with joint estimation of acoustic and excitation model parameters,” in *Proc. ISCA SSW7*, 2010, pp. 88–93.

[10] K. Nakamura, K. Hashimoto, Y. Nankaku, and K. Tokuda, “Integration of spectral feature extraction and modeling for HMM-based speech synthesis,” *IEICE Trans Inf. Syst.*, vol. 97, no. 6, pp. 1438–1448, 2014.

[11] K. Tokuda and H. Zen, “Directly modeling speech waveforms by neural networks for statistical parametric speech synthesis,” in *Proc. ICASSP*, 2015, pp. 4215–4219.

[12] K. Tokuda, T. Kobayashi, and S. Imai, “Adaptive cepstral analysis of speech,” *IEEE Trans. Speech Audio Process.*, vol. 3, no. 6, pp. 481–489, 1995.

[13] A.V. Oppenhem and R.W. Schafer, *Descrete-Time Signal Processing*, Prentice Hall, 1989.

[14] 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.

[15] “REAPER: Robust Epoch And Pitch EstimatoR,” `https://github.com/google/REAPER`, 2015.

[16] M. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q.-V. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, and G. Hinton, “On rectified linear units for speech processing,” in *Proc. ICASSP*, 2013, pp. 3517–3521.

[17] H. Sak, A. Senior, and F. Beaufays, “Long short-term memory recurrent neural network architectures for large scale acoustic modeling,” in *Proc. Interspeech*, 2014.

[18] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[19] J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization,” *The Journal of Machine Learning Research*, pp. 2121–2159, 2011.

[20] R. Williams and J. Peng, “An efficient gradient-based algorithm for on-line training of recurrent network trajectories,” *Neural Comput.*, vol. 2, no. 4, pp. 490–501, 1990.

[21] G. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” *arXiv preprent arXiv:1207.0580*, 2012.