Modulation spectrum-constrained trajectory error training for mixture density network-based speech synthesis

Sangjun Park, and Minsoo Hahn

Citation: The Journal of the Acoustical Society of America **144**, EL151 (2018); doi: 10.1121/1.5052206
View online: https://doi.org/10.1121/1.5052206
View Table of Contents: http://asa.scitation.org/toc/jas/144/3
Published by the Acoustical Society of America

Modulation spectrum-constrained trajectory error training for mixture density network-based speech synthesis

Sangjun Park and Minsoo Hahn[a)]
School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, Republic of Korea
psj@kaist.ac.kr, mshahn2@kaist.ac.kr

Abstract: In statistical parametric speech synthesis, a mixture density network is employed to address the limitations of a linear output layer such as pre-computed fixed variances and the unimodal assumption. However, it also has a defect, i.e., it cannot deploy a static-dynamic constraint needed in the training phase for high-quality speech synthesis. To cope with this problem, this paper proposes a training algorithm based on the minimum trajectory error for a mixture density network. And a modulation spectrum-constrained loss function is also proposed to alleviate the over-smoothing effect. The experimental results confirm meaningful improvement both in objective and subjective performance measures. [DDO]
Date Received: July 8, 2018 **Date Accepted:** August 14, 2018

1. Introduction

The speech quality of statistical parametric speech synthesis (SPSS) has been improved noticeably through the use of deep neural networks (DNNs),[1] which show better performance in representing the complex, nonlinear, and high-dimensional relationships between linguistic and acoustic features as compared to conventional hidden Markov models.[2] Over the last few years, several end-to-end speech synthesis frameworks generating human-like synthetic speech, including WaveNet, Tacotron, and Deepvoice, have been proposed.[3–5] However, they work so poorly under small corpus conditions and also incur high computation costs. Hence, SPSS approaches are still more useful in real environments.

In SPSS, several studies with various DNN-based architectures have been reported.[6–10] Usually they include a linear output layer and are trained with a mean squared error (MSE) loss function, generating acoustic features with a maximum likelihood parameter generation (MLPG) algorithm.[11] However, two unneglectable error sources still exist. They are the frame-wise independence assumption for the MSE criterion and the unimodal assumption for the linear output layer. Although the temporal information of speech is crucial for high-quality speech synthesis,[12] the MSE criterion fragments the relationship between the static and dynamic features in the training phase. Wu and King proposed the minimum trajectory error (MTE) criterion to moderate this drawback by adding a static-dynamic constraint on the MSE criterion.[13] The MTE training can generate a more natural trajectory, but it is still over-smoothed due to the linear output layer.[14] A mixture density network (MDN) can overcome this problem.[14–17] Multiple Gaussian mixtures in an MDN can represent the multimodality of speech and predict the variances while the linear output layer uses pre-computed fixed variances. Nevertheless, like the MSE criterion, the MDN also generates an unnatural trajectory because it cannot deploy temporal information during the training phase, i.e., the relationships between adjacent frames cannot be used. Another way of alleviating the over-smoothing effect is to utilize analytical features such as the global variance (GV) and the modulation spectrum (MS), known as perceptual cues.[18,19] Training/synthesis algorithms constrained on the GV and MS show improved clarity of synthetic speech.[20–23] However, these methods focus only on making the trajectory variation similar to the natural ones but do not consider the multimodality of speech.

In this paper, a novel MTE criterion-based training algorithm for MDNs is proposed to address both over-smoothing and unnatural trajectory problems. To introduce the MTE criterion, we reformulated the conventional iterative MLPG algorithm for MDNs (Ref. 11) into a closed-form solution utilizing only the most probable mixture (MPM) component. The proposed algorithm covers both the static-dynamic

[a)]Author to whom correspondence should be addressed.

constraint and the multimodality of speech and thus can generate more natural and clear synthetic speech. Furthermore, we introduce an MS constraint into the MTE loss function and mitigate the over-smoothing effect. Our experimental results confirm that the proposed algorithm improves the synthetic speech quality meaningfully both in objective and subjective evaluations.

2. Trajectory error training for an MDN constrained on a MS

2.1 Conventional training algorithm for a linear output layer

For a given frame-level linguistic feature sequence $\boldsymbol{X} = [\boldsymbol{x}_1^\top, \ldots, \boldsymbol{x}_t^\top, \boldsymbol{x}_T^\top]^\top$ and an observed acoustic sequence $\boldsymbol{O} = [\boldsymbol{o}_1^\top, \ldots, \boldsymbol{o}_t^\top, \boldsymbol{o}_T^\top]^\top$, the linear output layer in a DNN predicts the acoustic features directly[6–10] as $\hat{\boldsymbol{O}} = F_l(\boldsymbol{X}; \Phi_l)$, where $F_l(\cdot)$ is a DNN-based mapping function with a linear output layer and Φ_l denotes the parameters of $F_l(\cdot)$. Note that the acoustic feature sequence consists of the static and dynamic features as $\boldsymbol{o}_t = [\boldsymbol{c}_t^\top, \Delta\boldsymbol{c}_t^\top, \Delta^2\boldsymbol{c}_t^\top]^\top$, and $\boldsymbol{O}$ can be obtained as $\boldsymbol{O} = \boldsymbol{W}\boldsymbol{C}$, where $\boldsymbol{W}$ is a weight matrix for calculating dynamic features from static features.[11] To obtain $\hat{\boldsymbol{C}}$ from the observed acoustic features $\hat{\boldsymbol{O}}$ with the maximum likelihood criterion, an MLPG algorithm is used. It can be written as

$$\hat{\boldsymbol{C}} = (\boldsymbol{W}^\top \Sigma^{-1} \boldsymbol{W})^{-1} \boldsymbol{W}^\top \Sigma^{-1} \hat{\boldsymbol{O}}, \tag{1}$$

where $\Sigma = \mathrm{diag}[\Sigma_1, \ldots, \Sigma_t, \ldots, \Sigma_T]$ is a covariance matrix sequence.[11] Note that $F_l(\cdot)$ predicts only the mean of the acoustic features. For this reason, Σ is pre-computed over the training corpus. The MSE loss is designed to minimize the L2 loss of the observed acoustic features $\hat{\boldsymbol{O}}$, whereas the trajectory loss is defined as the L2 loss of the static acoustic features $\hat{\boldsymbol{C}}$ in Eq. (1). Now, the MSE and MTE loss functions can be written as follows:

$$L_{\mathrm{MSE}}(\boldsymbol{O}, \hat{\boldsymbol{O}}) = \frac{1}{T}(\hat{\boldsymbol{O}} - \boldsymbol{O})^\top (\hat{\boldsymbol{O}} - \boldsymbol{O}), \tag{2}$$

$$L_{\mathrm{MTE}}(\boldsymbol{C}, \hat{\boldsymbol{C}}) = \frac{1}{T}(\hat{\boldsymbol{C}} - \boldsymbol{C})^\top (\hat{\boldsymbol{C}} - \boldsymbol{C}). \tag{3}$$

The MTE training algorithm has a static-dynamic constraint based on the MLPG algorithm in the training phase and can generate a more natural synthetic speech.[13] However, this method still suffers from over-smoothing due to the unimodal assumption of the linear output layer.

2.2 Conventional training algorithm for an MDN

An MDN, which utilizes the multimodal feature distribution using a Gaussian mixture model (GMM), can overcome the above problem of a linear output layer. The MDN maps $\boldsymbol{X}$ to a GMM parameter sequence as $\Lambda = [\boldsymbol{\lambda}_1^\top, \ldots, \boldsymbol{\lambda}_t^\top, \ldots, \boldsymbol{\lambda}_T^\top]^\top = F_m(\boldsymbol{X}; \Phi_m)$,[14] where $\boldsymbol{\lambda}_t$ and $F_m(\cdot)$ denote the GMM parameters at the tth frame and the DNN-based mapping function with the MDN, respectively. $F_m(\cdot)$ can be trained by maximizing the GMM likelihood. The maximum likelihood (ML) loss function then becomes

$$L_{\mathrm{ML}}(\boldsymbol{O}, \boldsymbol{\Lambda}) = -\frac{1}{T}\sum_{t=1}^{T} \log \sum_{m=1}^{M} w_{m,t} \cdot N(\boldsymbol{o}_t; \boldsymbol{\mu}_{m,t}, \boldsymbol{\sigma}_{m,t}), \tag{4}$$

where M is the number of mixtures and $w_{m,t}$, $\boldsymbol{\mu}_{m,t}$, and $\boldsymbol{\sigma}_{m,t}$ correspond to the mixture weight, the mean, and the variance of the mth Gaussian component at the tth frame. From the GMM parameters $\boldsymbol{\Lambda}$, $\hat{\boldsymbol{C}}$ can also be obtained using an MLPG algorithm.[11] The MLPG algorithm for the MDN can be written as

$$\boldsymbol{W}^\top \overline{\Sigma^{-1}} \boldsymbol{W} \hat{\boldsymbol{C}} = \boldsymbol{W}^\top \overline{\Sigma^{-1}\boldsymbol{M}}, \tag{5a}$$

$$\overline{\Sigma^{-1}} = \mathrm{diag}\left[\overline{\Sigma_1^{-1}}, \ldots, \overline{\Sigma_T^{-1}}\right], \tag{5b}$$

$$\overline{\Sigma_t^{-1}} = \sum_{m=1}^{M} \gamma_t(m) \overline{\Sigma_{m,t}^{-1}}, \tag{5c}$$

$$\overline{\Sigma^{-1}\boldsymbol{M}} = \left[\overline{\Sigma_1^{-1}\boldsymbol{\mu}_1}^\top, \ldots, \overline{\Sigma_T^{-1}\boldsymbol{\mu}_T}^\top\right]^\top, \tag{5d}$$

$$\overline{\Sigma_t^{-1}\boldsymbol{\mu}_t} = \sum_{m=1}^{M} \gamma_t(m) \Sigma_{m,t}^{-1} \boldsymbol{\mu}_{m,t}, \tag{5e}$$

where $\gamma_t(m)$ is the occupancy probability, which can be solved with an iterative Expectation-Maximization (EM) algorithm. Due to the iterative-form solution, the trajectory loss of MDN cannot be derived, as opposed to Eq. (3).

2.3 Proposed trajectory error training algorithm for an MDN

To utilize the aforementioned advantages of both the MTE criterion and an MDN, we propose a new MTE training algorithm for an MDN. To derive the MTE loss function for an MDN, we reformulated Eq. (5a) into a closed-form solution using only the MPM component. This approach can be considered as a suboptimal solution. In this case, $\gamma_t(m)$ is 1 at every time step with $\boldsymbol{M} = 1$. Accordingly, Eqs. (5b) and (5d) can be rewritten as $\overline{\boldsymbol{\Sigma}^{-1}} = \boldsymbol{\Sigma}_{\mathrm{MPM}}^{-1} = \mathrm{diag}[\boldsymbol{\Sigma}_{m_{\mathrm{MPM}}^{(1)},1}^{-1}, \ldots, \boldsymbol{\Sigma}_{m_{\mathrm{MPM}}^{(T)},T}^{-1}]$ and $\overline{\boldsymbol{\Sigma}^{-1}\boldsymbol{M}} = \boldsymbol{\Sigma}_{\mathrm{MPM}}^{-1}\boldsymbol{M}_{\mathrm{MPM}} = [\overline{\boldsymbol{\Sigma}_{m_{\mathrm{MPM}}^{(1)},1}^{-1}\boldsymbol{\mu}_{m_{\mathrm{MPM}}^{(1)},1}}, \ldots, \overline{\boldsymbol{\Sigma}_{m_{\mathrm{MPM}}^{(T)},T}^{-1}\boldsymbol{\mu}_{m_{\mathrm{MPM}}^{(T)},T}}]^\top$, respectively, where $m_{\mathrm{MPM}}^{(t)}$ denotes the MPM at the tth frame. Finally, Eq. (5a) can be reformulated as follows:

$$\begin{aligned} \boldsymbol{W}^\top\boldsymbol{\Sigma}_{\mathrm{MPM}}^{-1}\boldsymbol{W}\hat{\boldsymbol{C}} &= \boldsymbol{W}^\top\boldsymbol{\Sigma}_{\mathrm{MPM}}^{-1}\boldsymbol{M}_{\mathrm{MPM}}, \\ \hat{\boldsymbol{C}} &= (\boldsymbol{W}^\top\boldsymbol{\Sigma}_{\mathrm{MPM}}^{-1}\boldsymbol{W})^{-1}\boldsymbol{W}^\top\boldsymbol{\Sigma}_{\mathrm{MPM}}^{-1}\boldsymbol{M}_{\mathrm{MPM}}. \end{aligned} \tag{6}$$

Using this closed-form solution, the MTE loss function for the MDN can be defined as follows:

$$\begin{aligned} L_{\mathrm{MTE2}}(\boldsymbol{C}, \hat{\boldsymbol{C}}) = \frac{1}{T}\Big((\boldsymbol{W}^\top\boldsymbol{\Sigma}_{\mathrm{MPM}}^{-1}\boldsymbol{W})^{-1}\boldsymbol{W}^\top\boldsymbol{\Sigma}_{\mathrm{MPM}}^{-1}\boldsymbol{M}_{\mathrm{MPM}} - \boldsymbol{C}\Big)^\top \\ \times \Big((\boldsymbol{W}^\top\boldsymbol{\Sigma}_{\mathrm{MPM}}^{-1}\boldsymbol{W})^{-1}\boldsymbol{W}^\top\boldsymbol{\Sigma}_{\mathrm{MPM}}^{-1}\boldsymbol{M}_{\mathrm{MPM}} - \boldsymbol{C}\Big). \end{aligned} \tag{7}$$

Note that $\boldsymbol{\Sigma}$ is also trainable, in contrast to Eq. (3). By the way, two ways to determine the MPM can be expressed as follows:

$$m_{\mathrm{MPM}}^{(t)} = \arg\max_m w_{m,t}, \tag{8}$$

$$m_{\mathrm{MPM}}^{(t)} = \arg\max_m N(\boldsymbol{o}_t; \boldsymbol{\mu}_{m,t}, \boldsymbol{\sigma}_{m,t}). \tag{9}$$

In the synthesis phase, the MPM is determined by Eq. (8) because $\boldsymbol{o}_t$ in Eq. (9) is unobservable. On the other hand, in order to train the mixtures corresponding to the training data, the MPM in Eq. (9) should be used in the training phase and it then can be considered as a teacher-forcing method.[24] If the MPM in Eq. (8) is adopted in the training phase, only one mixture is chosen as the MPM for the given linguistic feature, and the output of the DNN can be smoothed similarly to a linear output layer. In addition, when the MDN is trained by only the MTE loss function in Eq. (7), the parameters of the mixtures except for those of the MPM would be updated erroneously because they are affected by the shared layers. In order to train the MDN stably, we propose a new loss function to optimize Eqs. (4) and (7) jointly, as follows:

$$L_{\mathrm{MTE\,MDN}}(\boldsymbol{C}, \hat{\boldsymbol{C}}) = L_{ML}(\boldsymbol{W}\boldsymbol{C}, \boldsymbol{\Lambda}) + L_{\mathrm{MTE2}}(\boldsymbol{C}, \hat{\boldsymbol{C}}). \tag{10}$$

2.4 Proposed MS-constrained training algorithm

Although the training algorithm in Sec. 2.3 can generate a more natural trajectory due to the use of a static-dynamic constraint, the generated features can still be smoothed by outliers because they have not a multimodal but an irregular distribution. To address this problem, we propose an MS-constrained MTE loss function. It improves the synthetic speech quality by making the trajectory variation similar to natural ones. We defined the MS as a log power spectrum of the acoustic feature sequence[21] $\boldsymbol{S}(\boldsymbol{C}) = [\boldsymbol{S}_1, \ldots, \boldsymbol{S}_D]$, $\boldsymbol{S}_d = [\boldsymbol{S}_d^0, \ldots, \boldsymbol{S}_d^N]$, where D and N correspondingly denote the acoustic feature dimension and half of the fast Fourier transform (FFT) size. The segment-level MS, not the utterance-level MS, is used due to the normalization effect; it is computed regardless of the utterance length without zero-padding.[21] The MS loss function can be written as

$$L_{\mathrm{MS}}(\boldsymbol{C}, \hat{\boldsymbol{C}}) = \frac{1}{K}\big(\boldsymbol{S}(\hat{\boldsymbol{C}}) - \boldsymbol{S}(\boldsymbol{C})\big)^\top\big(\boldsymbol{S}(\hat{\boldsymbol{C}}) - \boldsymbol{S}(\boldsymbol{C})\big), \tag{11}$$

where K, $\boldsymbol{S}(\boldsymbol{C})$, and $\boldsymbol{S}(\hat{\boldsymbol{C}})$ are the number of MS segments and the MS of natural and generated acoustic features, respectively. From Eqs. (10) and (11), the MS-constrained MTE loss function for the MDN is defined as

Table 1. Acoustic model configurations for each experimental system.

Notation	Output layer	Training phase	Synthesis phase
MTE	Linear output layer	Update Φ_l with Eq. (3)	Generate $\hat{\boldsymbol{C}}$ with Eq. (1)
ML	MDN output layer	Update Φ_m with Eq. (4)	Generate $\hat{\boldsymbol{C}}$ with Eq. (5a)
Proposed-1	MDN output layer	Update Φ_m with Eqs. (9) and (10)	Generate $\hat{\boldsymbol{C}}$ with Eqs. (6) and (8)
Proposed-2	MDN output layer	Update Φ_m with Eqs. (9) and (13)	Generate $\hat{\boldsymbol{C}}$ with Eqs. (6) and (8)

$$L_{\text{MTE MDN MS}}(\boldsymbol{C},\hat{\boldsymbol{C}}) = L_{\text{MTE MDN}}(\boldsymbol{C},\hat{\boldsymbol{C}}) + L_{\text{MS}}(\boldsymbol{C},\hat{\boldsymbol{C}}). \tag{12}$$

In contrast to Eq. (10), each term in Eq. (12) has a different perspective on improving the speech quality; $L_{\text{MTE MDN}}(\boldsymbol{C},\hat{\boldsymbol{C}})$ and $L_{\text{MS}}(\boldsymbol{C},\hat{\boldsymbol{C}})$ involve the generation loss and variation of speech, respectively. If the momentum is biased for $L_{\text{MS}}(\boldsymbol{C},\hat{\boldsymbol{C}})$, the trained model focuses only on the high-frequency component of the features and not on the generation loss. This problem can be solved by adding an emphasis coefficient α, which is used in an MS-based postfilter.[21] The final loss function can be written as follows:

$$L_{\text{MTE MDN MS}}(\boldsymbol{C},\hat{\boldsymbol{C}}) = (1-\alpha)L_{\text{MTE MDN}}(\boldsymbol{C},\hat{\boldsymbol{C}}) + \alpha L_{\text{MS}}(\boldsymbol{C},\hat{\boldsymbol{C}}). \tag{13}$$

Hence, a larger α is for a more natural variation in speech, while a smaller α, for less generation loss in training.

3. Experiments and results

3.1 Experiment configuration

The Blizzard Challenge 2013 Nancy corpus was used as the training data. It consists of 12 095 English utterances (~18 h) sampled at 16 kHz.[25] 90% and 10% of the corpus except for 100 evaluation utterances were used as the training and validation data, respectively. The WORLD vocoder was utilized for the analysis/synthesis of the acoustic features.[26] The 25th order Mel-cepstrum coefficients (MCs), coded aperiodicity (AP), continuous *F*0 (Ref. 27), and voiced/unvoiced (V/UV) measures were extracted for each frame with a 5 ms shift. In addition, 372 linguistic features, e.g., the dot level, stress, inflection, playability, and the quin-Lessemes identity, were extracted based on Lessemes.[28]

To investigate the performance of the proposed algorithm, experimental systems were constructed by adopting a bi-directional long short-term memory (bLSTM) recurrent neural network for a duration model (DM) and an acoustic model (AM). Three bLSTM layers with 64 and 256 memory cells were used for the DM and the AM, respectively. The linear output layer trained with the MSE criterion by Eq. (2) was used in the DM. The detailed AM configurations are given in Table 1. Empirically determined different numbers of mixtures for each acoustic feature were used; four for MC, two for AP, two for *F*0, and one for V/UV. The MS is computed from the acoustic feature sequence but without V/UV. For segment-level MS computations, a Bartlett window with a frame-length of 25 and with a frame shift size of 12 was used, and the FFT size was 64 frames. The emphasis coefficient α of Proposed-2 was determined empirically through listening tests. We set α to 0.2, because the MTE loss increases rapidly when $\alpha \geq 0.3$ and we also found some unstable fluctuations for some samples when $\alpha = 0.3$. All systems were trained using a mini-batch stochastic gradient descent-based backpropagation algorithm with the Adam optimizer.[29] The early stopping method was adopted to determine the number of training epochs.[30] All systems were implemented in PyTorch.[31]

For a subjective evaluation, preference tests of four cases were conducted; 20 listeners assessed 20 synthetic speech pairs for each case shown in Table 1; the Mel-cepstral distortion (MCD) and root-mean-square error (RMSE) of *F*0 were computed

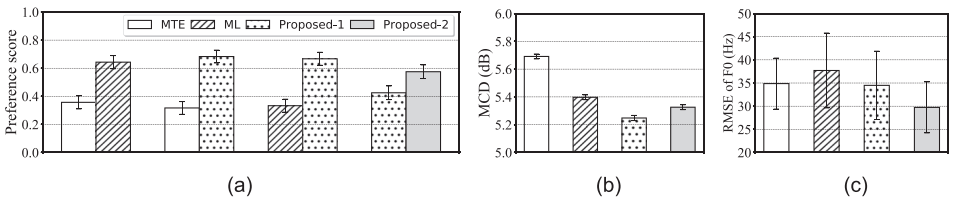

Fig. 1. Subjective and objective evaluation results with a 95% confidence interval: (a) preference score for four cases: MTE vs ML, MTE vs Proposed-1, ML vs Proposed-1 and Proposed-1 vs Proposed-2 in order; (b) MCD; and (c) RMSE of *F*0.

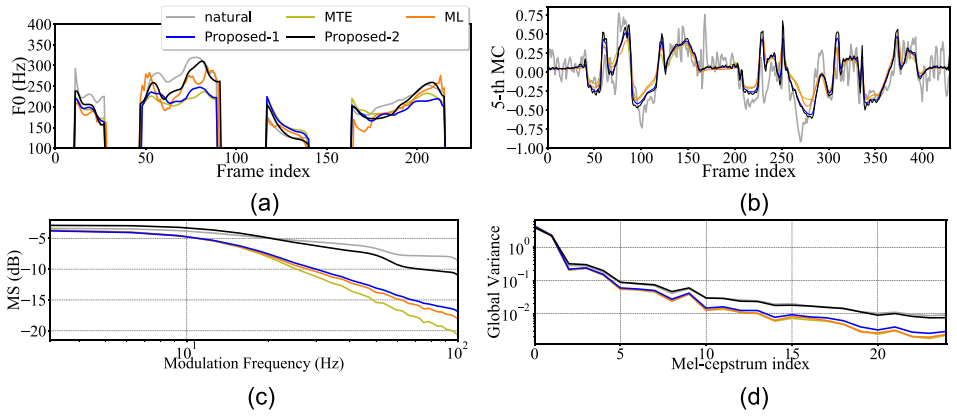

Fig. 2. (Color online) Comparative plot of the generated features with the natural features in *F*0, MC, MS, and GV-domains: (a) *F*0 contour example; (b) fifth MC contour example; (c) averaged MS of the 20th MC sequences; and (d) averaged GV of MC sequences.

for an objective evaluation. The speech samples were generated from the DM and the AM for the subjective test, whereas they were generated from only the AMs for the objective test. All experimental results were drawn for 100 evaluation utterances.

3.2 Results and discussion

Figure 1 summarizes our subjective and objective test results. Figure 2 shows the generated feature plot with its natural counterpart. ML and the proposed methods are preferred to MTE, as shown in Fig. 1(a). Although MTE predicted *F*0 fairly well by deploying temporal information, Fig. 1(b) shows that it is limited when used to model the MC multimodality, as described in earlier work.[15] Figure 2(b) also confirms that the generated MC sequence from MTE is smoother than the others.

Proposed-1 outperforms ML on the objective and subjective measures. Specifically, *F*0 generated from ML is degraded by abrupt fluctuations in time, as illustrated in Figs. 1(c) and 2(a). This phenomenon also arose in earlier work,[17] having been caused by the absence of dynamic statistics in the training procedure. By introducing the MTE criterion, Proposed-1 moderated this drawback of the ML criterion. However, the features generated by Proposed-1 are still smoothed by the outliers.

By introducing an MS constraint, Proposed-2 improves this smoothing problem with the better MS and GV, which are more similar to the natural ones, as shown in Figs. 2(c) and 2(d). Note that the lower MS and GV mean a more smoothed and degraded trajectory. A significant improvement in the *F*0 estimation was achieved with minor degradation in the MC estimation, as depicted in Figs. 1(b) and 1(c). We can therefore conclude that certain amounts of natural fluctuations of *F*0 and the MC in the time domain improve the naturalness and clarity of synthetic speech despite the higher generation loss. Our experimental result also supports this idea. Whereas the MC values generated by other algorithms were narrowly distributed, those by Proposed-2 were rather widely distributed like that of the natural samples.

4. Conclusion

In this paper, a novel training algorithm for an MDN based on the MTE criterion was proposed in an attempt to overcome the quality degradation of synthetic speech caused by the frame-wise independence and unimodal assumptions of conventional DNN-based SPSS algorithms.

Our proposed algorithm improved the naturalness and the clarity of synthetic speech with more than a 60% preference score in a subjective evaluation when compared to the conventional algorithms and more precise *F*0 and unsmoothed MC in an objective evaluation. We also investigated an MS constraint which reduces over-smoothing caused by outliers and got a reasonably successful improvement. Furthermore, the proposed parameter generation algorithm in a closed-form can save the computation cost of the iterative MLPG algorithm. Considering all these results, we can say that our proposed methods achieved a meaningful improvement in the synthetic speech quality.

Acknowledgments

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Science, ICT, and Future Planning (Grant No. 2017R1A2B4011357).

References and links

[1]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 Process. Mag. **32**(3), 35–52 (2015).

[2]K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura, "Speech synthesis based on hidden Markov models," Proc. IEEE **101**(5), 1234–1252 (2013).

[3]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," CoRR abs/1609.03499 (2016), arxiv.org/abs/1609.03499.

[4]J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. J. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning Wavenet on mel spectrogram predictions," in *Proceedings of the 2018 IEEE ICASSP*, Calgary, Canada (2018), pp. 4779–4783.

[5]W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: Scaling text-to-speech with convolutional sequence learning," in *Proceedings of the International Conference on Learning Representations (ICLR 2018)*, Vancouver, Canada (2018).

[6]H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proceedings of the 2013 IEEE ICASSP*, Vancouver, Canada (2013), pp. 7962–7966.

[7]Z. H. Ling, L. Deng, and D. Yu, "Modeling spectral envelopes using restricted Boltzmann machines and deep belief networks for statistical parametric speech synthesis," IEEE Trans. Audio, Speech, Lang. Process. **21**(10), 2129–2139 (2013).

[8]H. Zen and H. Sak, "Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis," in *Proceedings of the 2015 IEEE ICASSP*, Brisbane, Australia (2015), pp. 4470–4474.

[9]Z. Wu, C. Valentini-Botinhao, O. Watts, and S. King, "Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis," in *Proceedings of the 2015 IEEE ICASSP*, Brisbane, Australia (2015), pp. 4460–4464.

[10]H. Zen, Y. Agiomyrgiannakis, N. Egberts, F. Henderson, and P. Szczepaniak, "Fast, compact, and high quality LSTM-RNN based statistical parametric speech synthesizer for mobile devices," in *Proceedings of INTERSPEECH 2016*, San Francisco, CA (2016), pp. 2273–2277.

[11]K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proceedings of the 2000 IEEE ICASSP* (2000), Vol. 3, pp. 1315–1318.

[12]L. Xu and Y. Zheng, "Spectral and temporal cues for phoneme recognition in noise," J. Acoust. Soc. Am **122**(3), 1758–1764 (2007).

[13]Z. Wu and S. King, "Improving trajectory modelling for DNN-based speech synthesis by using stacked bottleneck features and minimum generation error training," IEEE/ACM Trans. Audio, Speech, Lang. Process. **24**(7), 1255–1265 (2016).

[14]C. M. Bishop, "Mixture density networks," Technical Report, Birmingham, AL, publications.aston.ac.uk/373/.

[15]H. Zen and A. Senior, "Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis," in *Proceedings of the 2014 IEEE ICASSP*, Florence, Italy (2014), pp. 3844–3848.

[16]K. Richmond, "Trajectory mixture density networks with multiple mixtures for acoustic-articulatory inversion," in *Advances in Nonlinear Speech Processing* (Springer, Berlin, Heidelberg, 2007), pp. 263–272.

[17]X. Wang, S. Takaki, and J. Yamagishi, "An autoregressive recurrent mixture density network for parametric speech synthesis," in *Proceedings of the 2017 IEEE ICASSP*, New Orleans, LA (2017), pp. 4895–4899.

[18]V. Tyagi, I. McCowan, H. Misra, and H. Bourlard, "Mel-cepstrum modulation spectrum (mcms) features for robust ASR," in *2003 IEEE Workshop on ASRU* (2003), pp. 399–404.

[19]R. Drullman, J. M. Festen, and R. Plomp, "Effect of reducing slow temporal modulations on speech reception," J. Acoust. Soc. Am **95**(5), 2670–2680 (1994).

[20]T. Toda and K. Tokuda, "A speech parameter generation algorithm considering global variance for HMM-based speech synthesis," IEICE Trans. Inf. Syst. **E90-D**(5), 816–824 (2007).

[21]S. Takamichi, T. Toda, A. W. Black, G. Neubig, S. Sakti, and S. Nakamura, "Postfilters to modify the modulation spectrum for statistical parametric speech synthesis," IEEE Trans. Audio, Speech, Lang. Process. **24**(4), 755–767 (2016).

[22]S. Takamichi, T. Toda, A. W. Black, and S. Nakamura, "Modulation spectrum-constrained trajectory training algorithm for GMM-based voice conversion," in *Proceedings of the 2015 IEEE ICASSP*, Brisbane, Australia (2015), pp. 4859–4863.

[23]K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Trajectory training considering global variance for speech synthesis based on neural networks," in *Proceedings of the 2016 IEEE ICASSP*, Shanghai, China (2016), pp. 5600–5604.

[24]R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," Neural Comput. **1**(2), 270–280 (1989).

[25]S. King, K. T. A. W. Black, and K. Prahallad, "The blizzard challenge 2013," in *Proceedings of the Blizzard Challenge 2013*, Barcelona, Spain (2013), pp. 1–10.

[26]M. Morise, F. Yokomori, and K. Ozawa, "WORLD: A vocoder-based high-quality speech synthesis system for real-time applications," IEICE Trans. Inf. Syst. **E99-D**(7), 1877–1884 (2016).

[27]K. Yu and S. Young, "Continuous F0 modeling for HMM-based statistical parametric speech synthesis," IEEE Trans. Audio, Speech, Lang. Process. **19**(5), 1071–1079 (2011).

[28]M. Munro, S. Turner, A. Munro, and K. Campbell, "Use of Lessemes in text-to-speech synthesis," in *Collective Writings on the Lessac Voice and Body Work: A Festschrift* (Llumina Press, Coral Springs, FL, 2009), pp. 362–374.
[29]D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," CoRR abs/1412.6980 (2014), arxiv.org/abs/1412.6980.
[30]R. Caruana, S. Lawrence, and L. Giles, "Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping," in *Advances in Neural Information Processing Systems* (MIT Press, Cambridge, MA, 2001), pp. 402–408.
[31]A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *Proceedings of the NIPS Autodiff Workshop* (2017).