Sampling-based speech parameter generation
using moment-matching networks

Shinnosuke Takamichi [1], *Tomoki Koriyama* [2], *Hiroshi Saruwatari* [1]

[1] Graduate School of Information Science and Technology, The University of Tokyo, Japan.
[2] Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology, Japan.

{shinnosuke_takamichi,hiroshi_saruwatari}@ipc.i.u-tokyo.ac.jp, koriyama@ip.titech.ac.jp
email@address

Abstract

This paper presents sampling-based speech parameter generation using moment-matching networks for Deep Neural Network (DNN)-based speech synthesis. Although people never produce exactly the same speech even if we try to express the same linguistic and para-linguistic information, typical statistical speech synthesis produces completely the same speech, i.e., there is no inter-utterance variation in synthetic speech. To give synthetic speech natural inter-utterance variation, this paper builds DNN acoustic models that make it possible to randomly sample speech parameters. The DNNs are trained so that they make the moments of generated speech parameters close to those of natural speech parameters. Since the variation of speech parameters is compressed into a low-dimensional simple prior noise vector, our algorithm has lower computation cost than direct sampling of speech parameters. As the first step towards generating synthetic speech that has natural inter-utterance variation, this paper investigates whether or not the proposed sampling-based generation deteriorates synthetic speech quality. In evaluation, we compare speech quality of conventional maximum likelihood-based generation and proposed sampling-based generation. The result demonstrates the proposed generation causes no degradation in speech quality.

Index Terms: deep neural network, DNN-based speech synthesis, moment-matching network, sampling-based speech parameter generation

1. Introduction

Statistical parametric speech synthesis [1] is a method of synthesizing speech using statistical models. One of the final goals of speech synthesis is to synthesize speech as *natural* as humans' speech. Speech quality is an important factor of naturalness, and various methods have been developed for improving the quality of synthesized speech [2, 3, 4]. In particular, Deep Neural Network (DNN)-based speech synthesis [5] has remarkably improved the quality of synthesized speech. However, speech quality is just one factor of naturalness, so the *naturalness* of synthetic speech needs to be evaluated from other perspectives.

This paper focuses on a new metric for naturalness: inter-utterance variation within a given context [6]. As shown in Fig. 1, conventional DNN-based speech synthesis produces speech on the basis of a minimum mean error criterion, and generates the same synthetic speech from the given context. Therefore, when the context is fixed, the synthesized speech is always the same and the repeatedly synthesized speech is just recorded and played back. On the other hand, people
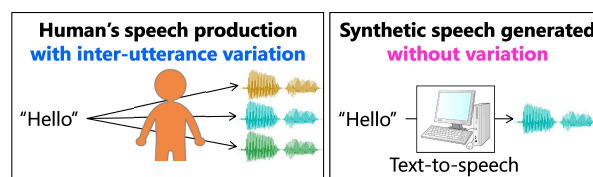

Figure 1: *Comparison of natural and synthetic speech. When input context is fixed, humans' speech has variation between utterances, but conventional synthetic speech does not.*

never produce exactly the same speech even if we try to fix linguistic and para-linguistic information. We call this variation inter-utterance variation (or intra-speaker variation) [6], and this paper addresses ways to synthesize speech that has inter-utterance variation. The straightforward way is to explicitly model the inter-utterance variation using repeatedly spoken speech data that have the same context. However, typical training data for speech synthesis do not include such data, and collecting the data is time-consuming work. Another way is to randomly sample speech from the conditional probability distribution of speech parameters. Shannon et al. [7] evaluated random sampling from the trajectory Hidden Markov Models (HMMs) [8]. They reported that speech quality in randomly-sampled synthetic speech is significantly degraded below that in Maximum Likelihood (ML)-based synthetic speech. One reason for quality degradation is time quantization by HMMs [9], and sampling from trajectory DNN [10], mixture density networks [11], or trajectory-RNADE (Real-valued Neural Autoregressive Density-Estimator) [12] is the better solution. However, in practice, sampling from such distributions is computationally inefficient since a speech parameter sequence is a high dimensional feature.

This paper proposes sampling-based speech parameter generation using moment-matching networks. DNNs are trained to match moments of natural speech parameters and synthetic speech parameters. The variation of speech parameters is compressed into a low-dimensional simple prior noise vector, and the DNNs transform the simple noise into the speech parameter variation. In synthesis, the synthetic speech parameters are randomly sampled using the sampled prior noise. Since the variation is represented as simple noise, the proposed sampling-based generation is more computationally efficient than the direct sampling of speech parameters. As the first step towards generating synthetic speech that has natural inter-utterance variation, this paper investigates whether or not

the proposed sampling-based generation deteriorates synthetic speech quality. In evaluation, we compared speech quality of ML-based generation and proposed sampling-based generation, and the results demonstrate that the proposed generation causes no degradation in speech quality.

2. Conventional DNN-based speech synthesis and sampling from acoustic models

In conventional DNN-based speech synthesis, the DNN parameters (e.g., matrices and biases) are estimated to minimize the mean squared error between natural speech parameters and generated speech parameters [5, 13, 14]. The training criterion is equivalent to ML-based training with an isotropic Gaussian distribution (Gaussian distribution having an isotropic covariance matrix) and estimates only mean vectors of the Gaussian distribution. In synthesis, the same as in training, the acoustic models generate ML estimates of speech parameters given the input context. Therefore, when the input context is fixed, the generated speech parameters are always the same.

We can randomly sample speech parameters from more proper distributions such as trajectory models [7, 10]. Although the trajectory models make it possible to sample speech parameters considering static and delta constraints, the computation cost for sampling is too high because the trajectory model has large full-covariance matrices. Also, mixture density networks [11] can represent proper distributions such as Gaussian mixture models but such complicated models suffer from high computation cost and low accuracy of sampling[1].

3. Sampling-based speech parameter generation using moment-matching networks

This section describes how to randomly sample speech parameters from moment-matching networks. The training criterion does not assume an explicit distribution form (such as Gaussian [10] and mixture models [11]) but minimizes the distance between moments of training and generated data.

3.1. Moment-matching networks

3.1.1. Minimizing Maximum Mean Discrepancy (MMD) [16]

Let $\boldsymbol{y} = \left[\boldsymbol{y}_1^\top, \cdots, \boldsymbol{y}_t^\top, \cdots, \boldsymbol{y}_T^\top\right]^\top$ and $\hat{\boldsymbol{y}} = \left[\hat{\boldsymbol{y}}_1^\top, \cdots, \hat{\boldsymbol{y}}_t^\top, \cdots, \hat{\boldsymbol{y}}_T^\top\right]^\top$ be a feature sequence of training data and that generated through DNNs. $\boldsymbol{y}_t$ and $\hat{\boldsymbol{y}}_t$ are training and generated data at frame t. T is the sequence length. The DNNs that can sample $\hat{\boldsymbol{y}}$ are trained to minimize squared error between moments of $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$. Using a kernel trick, the criterion is represented using gram matrices, and is known as the square of (kernelized) Maximum Mean Discrepancy (MMD) as follows:

$$L_{\mathrm{MMD}}\left(\boldsymbol{y}, \hat{\boldsymbol{y}}\right) = \frac{1}{T^2}\left\{\mathrm{tr}\left(\mathbf{1}_T \cdot \boldsymbol{K}_y\left(\boldsymbol{y}, \boldsymbol{y}\right)\right) + \mathrm{tr}\left(\mathbf{1}_T \cdot \boldsymbol{K}_y\left(\hat{\boldsymbol{y}}, \hat{\boldsymbol{y}}\right)\right) - 2 \cdot \mathrm{tr}\left(\mathbf{1}_T \cdot \boldsymbol{K}_y\left(\boldsymbol{y}, \hat{\boldsymbol{y}}\right)\right)\right\}, \tag{1}$$

where $\mathrm{tr}\left(\cdot\right)$ denotes matrix trace. $\mathbf{1}_T$ is T-by-T matrix all components of which are 1. $\boldsymbol{K}_y\left(\boldsymbol{y}, \hat{\boldsymbol{y}}\right)$ is a gram matrix between $\boldsymbol{y}$

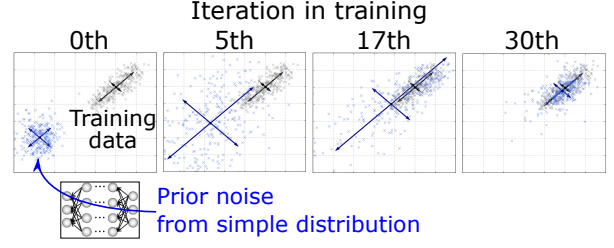[1]If the covariance matrix is diagonal, the computation cost can be reduced [15].

Figure 2: *Training of moment-matching networks. Distributions of training (black) and generated (blue) data are drawn with their 1st and 2nd moments. Networks are trained to transform prior simple noise into data distribution.*

and $\hat{\boldsymbol{y}}$. $\{t, \tau\}$-th component of $\boldsymbol{K}_y\left(\boldsymbol{y}, \hat{\boldsymbol{y}}\right)$ is an arbitrary kernel function between $\boldsymbol{y}_t$ and $\hat{\boldsymbol{y}}_\tau$. If the Gaussian kernel is chosen, 0th-through-infinite order moments are considered in training. The criterion L_{MMD} is used to train DNNs that have a low-dimensional noise vector $\boldsymbol{n}$ as input and $\hat{\boldsymbol{y}}$ as output. The DNNs can be regarded as models to transform statistics of $\boldsymbol{n}$ into those of $\boldsymbol{y}$ as shown in Fig. 2.

3.1.2. Minimizing conditional MMD [17]

The method in Section 3.1.1 can be extended to conditional distributions. Given the referred to input sequence $\boldsymbol{x} = \left[\boldsymbol{x}_1^\top, \cdots, \boldsymbol{x}_t^\top, \cdots, \boldsymbol{x}_T^\top\right]^\top$, the DNNs are trained to match moments of $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$. The input of DNNs is a concatenated vector $\tilde{\boldsymbol{x}} = [\boldsymbol{x}^\top, \boldsymbol{n}^\top]^\top$, and the following loss function called conditional MMD is used to train DNNs,

$$L_{\mathrm{CMMD}}\left(\tilde{\boldsymbol{x}}, \boldsymbol{y}, \hat{\boldsymbol{y}}\right) = \frac{1}{T^2}\left\{\mathrm{tr}\left(\boldsymbol{G}\left(\tilde{\boldsymbol{x}}\right) \cdot \boldsymbol{K}_y\left(\boldsymbol{y}, \boldsymbol{y}\right)\right)\right.$$
$$+ \mathrm{tr}\left(\boldsymbol{G}\left(\tilde{\boldsymbol{x}}\right) \cdot \boldsymbol{K}_y\left(\hat{\boldsymbol{y}}, \hat{\boldsymbol{y}}\right)\right)$$
$$\left. - 2 \cdot \mathrm{tr}\left(\boldsymbol{G}\left(\tilde{\boldsymbol{x}}\right) \cdot \boldsymbol{K}_y\left(\boldsymbol{y}, \hat{\boldsymbol{y}}\right)\right)\right\}, \tag{2}$$

$$\boldsymbol{G}\left(\tilde{\boldsymbol{x}}\right) = \tilde{\boldsymbol{K}}_x^{-1}\left(\tilde{\boldsymbol{x}}\right) \boldsymbol{K}_x\left(\tilde{\boldsymbol{x}}\right) \tilde{\boldsymbol{K}}_x^{-1}\left(\tilde{\boldsymbol{x}}\right), \tag{3}$$

$$\tilde{\boldsymbol{K}}_x\left(\tilde{\boldsymbol{x}}\right) = \boldsymbol{K}_x\left(\tilde{\boldsymbol{x}}\right) + \lambda \boldsymbol{I}_T, \tag{4}$$

where $\boldsymbol{I}_T$ is the T-by-T identical matrix, λ is the regularization coefficient, and $\boldsymbol{K}_x\left(\tilde{\boldsymbol{x}}\right)$ is a gram matrix of $\tilde{\boldsymbol{x}}$.

In generation, $\hat{\boldsymbol{y}}$ is randomly sampled from DNNs given the referred input $\boldsymbol{x}$ and randomly sampled $\boldsymbol{n}$.

3.2. Applying moment-matching networks to sampling-based speech parameter generation

Here, we describe how to perform sampling-based speech parameter generation using moment-matching networks. As shown in Fig. 3, DNNs are trained to minimize conditional MMD. $\boldsymbol{x}$ and $\boldsymbol{y}$ are the contextual feature sequence of input text and parameter sequence of output speech, respectively. $\boldsymbol{n}$ is sampled frame-by-frame from the simple distribution. The DNNs predict static and dynamic features of output speech parameters, and $\hat{\boldsymbol{y}}$ is determined considering these features [18]. In synthesis, after linguistic features are determined and the prior noise is sampled, the generation process is performed in the standard manner [5].

Since most of contextual factors are 1-of-K hot vectors and their combination becomes an enormous number, the kernel function between contextual factors does not have much
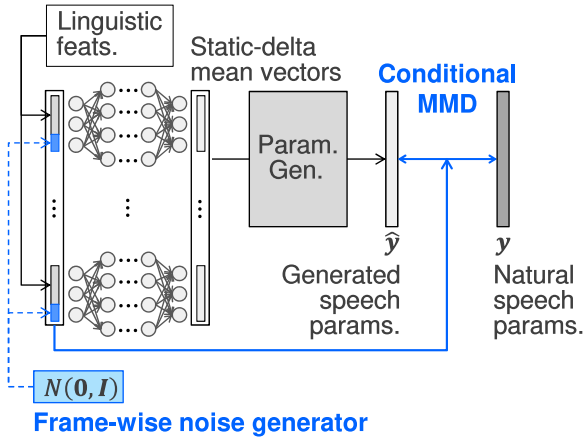
Figure 3: *Sampling-based speech parameter generation using moment-matching networks. Note that linguistic features are directly used in this figure for clear illustration, but bottleneck features [19] are used in place of linguistic features in actual implementation.*

Figure 4: *An example of generated speech parameter trajectories. We sampled five trajectories using proposed method. Conventional one is trained using mean squared error.*

information. Hence, we used bottleneck features instead of contextual features for kernel calculation. Other Feed-Forward networks that predict y from x are constructed using a mean squared error criterion [5], and continuous values of the specific hidden layer are used to calculate kernels.

3.3. Discussion

Since conditional MMD is a non-parametric measure, our algorithm can model a more complicated distribution than mixture density networks [11] or trajectory DNNs [10]. Additionally, since the variation of speech parameters is compressed into the low-dimensional prior noise vector, our sampling-based generation is more computationally efficient than conventional models [11, 10].

A Generative Adversarial Network (GAN) [20] and conditional GAN [21] are also non-parametric modeling of probability distributions. The GAN requires many tricky techniques to optimize since the optimization is a minimax problem [22]. We have already developed speech synthesis integrating GAN [4], but the algorithms presented in this paper is easier to optimize than the GAN-based methods. This is because our algorithm is just a minimization problem of conditional MMD. Here, we further discuss the relationships among GAN, our algorithm (moment matching), and conventional speech-related techniques. GAN minimizes divergence between natural and generation distribution, such as Jensen-Shannon divergence [20] and f-divergence [23], therefore, GAN-based methods are related to divergence-based speech processing such as β-divergence [24, 25]. On the other hand, moment matching explicitly uses the distance between moments, so our method is related to conventional speech processing methods using higher-order statistics, such as kurtosis [26], global variance [27] and modulation spectrum [3].

A conventional approach to add inter-utterance variation is to use of sentence-level context [28]. Whereas this method corresponds to adding speech expressions intentionally added by the speaker, our method adds speech randomness that is not intended by the speaker.

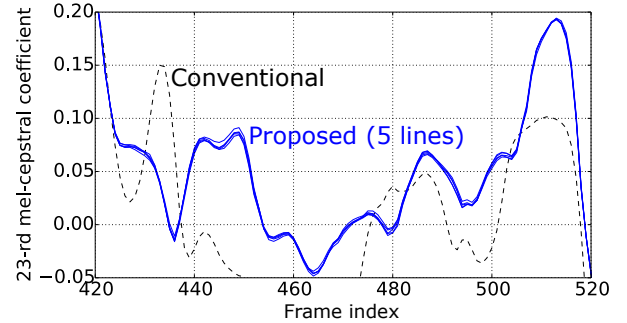Finally, we discuss whether our algorithm can deceive Anti-

Spoofing Verification (ASV) [29] which detects voice spoofing attacks. Deceiving the ASV has become the metric to evaluate speech synthesis [4], and one ASV technique is replay-attack detection [30] which determines whether the presented voice is raw or recorded. The technique detects recorded voices on the basis of degree of coincidence between the pre-recorded and presented voices. Conventional speech synthesis is easily detected by the replay-attack detection because repeatedly synthesized speech is completely the same as the recorded voice. On the other hand, we expect that our algorithm can mitigate the detection rate because it has inter-utterance variation.

4. Experimental evaluation

4.1. Experimental conditions

We used speech data of five female speakers taken from the ATR Japanese speech database [31]. Each speaker uttered 503 phonetically balance sentences. We used 450 sentences per speaker (2250 sentences in total) for the training and 53 sentences of one speaker for the evaluation. Speech signals were sampled at a rate of 16 kHz, and the shift length was set to 5 ms. The 0th-through-24th mel-cepstral coefficients were used as a spectral parameter and F_0 and 5 band-aperiodicity [32, 33] were used as excitation parameters. The WORLD analysis-synthesis system [34] was used to extract the parameters and synthesize the waveform. To improve training accuracy, speech parameter trajectory smoothing [35] with a 50 Hz cutoff modulation frequency was applied to the spectral parameters in the training data. Contextual features includes 274-dimensional linguistic features[2] (phonemes, mora position, and so on.) and 5-dimensional speaker codes [36]. The contextual features and speech features are normalized to have zero-mean unit-variance, and 80% of the silence frames were removed from the training data. The DNN architectures was Feed-Forward networks that includes 3×512-unit Rectified Linear Unit (ReLU) [37] hidden layers and a 75-unit linear output layer. The DNN predicted only spectral parameters, and F_0 and band aperiodicity of natural speech were used for waveform synthesis. The input of DNNs was 128-dimensional bottleneck features and a 3-dimensional noise vector per frame. The noise is sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I}_3)$ The regularization coefficient λ was set to 0.01. A Gaussian kernel was used as a kernel function for speech parameters y, i.e., $\exp\{-\|y_t - \hat{y}_\tau\|^2/\sigma^2\}$. σ was set so that

---

[2]We only used some of the prosody-related features because we predicted only the spectral parameters in this experiment.
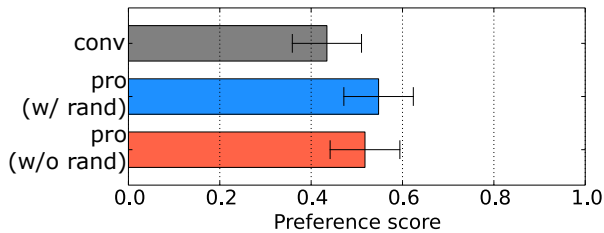
Figure 5: *Preference scores on speech quality with 95% confidence interval.*

$\|\boldsymbol{y}_t - \hat{\boldsymbol{y}}_\tau\|^2 < 1$ for all the training data [17]. The same kernel settings were used for the input features.

As described in Section 1, investigating naturalness of interutterance variation is our next work, and this paper investigates the effect of sampling-based generation in speech quality [7]. We evaluated the following speech synthesis systems:

conv: conventional speech synthesis minimizing mean squared error [5]

pro (w/ rand): proposed speech synthesis with sampling-based generation

pro (w/o rand): proposed speech synthesis with ML-based generation

For "pro (w/o rand)," we fixed the prior noise vector to **0** (ML estimates) in synthesis. "pro (w/o rand)" is trained the same as "pro (w/o rand)," but the speech parameters have no interutterance variation. Note that we did not applied emphasis algorithms such as cepstrum emphasis [38], global variance [39], and modulation spectrum [3, 40].

A preference AB test was conducted by eight listeners. We presented every pair of generated speech of three systems in random order, and forced listeners to select the speech sample that sounds better quality.

4.2. Experimental results

Fig. 4 shows an example of speech parameter sequence generated by conventional and proposed algorithms. We can see that the sequences of the proposed algorithm are significantly different from that of the conventional algorithm, and they have an inter-utterance variation.

Fig. 5 shows the results of the preference test on speech quality. We can see that sampling-based generation does not degrade speech quality below that of ML-based generation. Also, speech quality of our systems with non-parametric modeling is better than that of conventional speech synthesis assuming an isotropic Gaussian distribution. These results demonstrate that the proposed sampling-based generation can generate speech that has inter-utterance variation while preserving speech quality. Our next work is to compare the variation of natural and synthetic speech, and to apply the algorithm to end-to-end speech synthesis.

5. Conclusion

This paper presented sampling-based speech parameter generation using moment-matching networks to give synthetic speech natural inter-utterance variation. Although humans' speech has variation within the same linguistic and para-linguistic information, conventional speech synthesis always produces completely the same speech within the same context. To address this gap, we have proposed sampling-based generation using moment-matching networks. The neural networks were trained to transform the simple prior noise to the distribution (variation) of speech parameters. Since the training criterion (conditional maximum mean discrepancy) is non-parametric, our algorithm can model more complicated distributions than conventional mean squared error training assuming an isotropic Gaussian distribution. Furthermore, since the variation is compressed into the simple prior noise, the proposed sampling-based generation is more computationally efficient than direct sampling from distributions of speech parameters. We have investigated whether or not the proposed generation degrades synthetic speech quality. The results demonstrated that our sampling-based generation causes no degradation compared with maximum likelihood-based generation. As future work, We will investigate what degree the proposed algorithm reproduces natural inter-utterance variation.

6. Acknowledgements

7. References

[1] H. Zen, K. Tokuda, and A. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[2] S. Takamichi, T. Toda, Y. Shiga, S. Sakti, G. Neubig, and S. Nakamura, "Parameter generation methods with rich context models for high-quality and flexible text-to-speech synthesis," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 2, pp. 239–250, 2014.

[3] 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 Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 4, pp. 755–767, 2016.

[4] Y. Saito, S. Takamichi, and H. Saruwatari, "Training algorithm to deceive anti-spoofing verification for DNN-based speech synthesis," in *Proc. ICASSP*, Orleans, U.S.A., Mar. 2017.

[5] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. ICASSP*, Vancouver, Canada, May 2013.

[6] T. Inukai, T. Toda, G. Neubig, S. Sakti, and S. Nakamura, "Investigation of intra-speaker spectral parameter variation and its prediction towards improvement of spectral conversion metric," in *Proc. SSW8*, Barcelona, Spain, Aug. 2013, pp. 89–94.

[7] M. Shannon, H. Zen, and W. Byrne, "The effect of using normalized models in statistical speech synthesis," in *Proc. INTERSPEECH*, Florence, Italy, Jul. 2011, pp. 121–124.

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

[9] O. Watts, G. E. Henter, T. Merritt, Z. Wu, and S. King, "From HMMs to DNNs: where do the improvements come from?" in *Proc. ICASSP*, Shanghai, China, Mar. 2016, pp. 5505–5509.

[10] K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "The effect of neural networks in statistical parametric speech synthesis," in *Proc. ICASSP*, Brisbane, Australia, Apr. 2015, pp. 4455–4459.

[11] H. Zen and A. Senior, "Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis," in *Proc. ICASSP*, Florence, Italy, May 2014, pp. 3872–3876.

[12] B. Urial, I. M. S. Renals, C. V.-Botinhao, and J. Bridle, "Modelling acoustic feature dependencies with artificial neural networks: Trajectory-RNADE," in *Proc. ICASSP*, Brisbane, Australia, Apr. 2015, pp. 4465–4469.

[13] Z. Wu and S. King, "Minimum trajectory error training for deep neural networks, combined with stacked bottleneck features," in *Proc. INTERSPEECH*, Dresden, Germany, Sep. 2015, pp. 309–313.

[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*, Brisbane, Australia, Apr. 2015, pp. 4470–4474.

[15] K. Tokuda, H. Zen, and T. Kitamura, "Reformulating the HMM as a trajectory model," in *Beyond HMM Workshop on statistical modeling approach for speech recognition*, Kyoto, Japan, Dec. 2004.

[16] Y. Li, K. Swersky, and R. Zemel, "Generative moment matching networks," in *Proc. ICML*, Lille, France, Jul. 2015, pp. 1718–1727.

[17] Y. Ren, J. Li, Y. Luo, and J. Zhu, "Conditional generative moment-matching networks," in *Proc. NIPS*, Barcelona, Spain, Dec. 2016, pp. 2928–2936.

[18] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. ICASSP*, Istanbul, Turkey, June 2000, pp. 1315–1318.

[19] 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 *Proc. ICASSP*, Brisbane, Australia, Apr. 2015, pp. 4460–4464.

[20] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. WardeFarley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Proc. NIPS*, pp. 2672–2680, 2014.

[21] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv:1411.1784*, 2015.

[22] I. Goodfellow, "NIPS 2016 tutorial: Generative adversarial networks," in *Proc. NIPS*, Barcelona, Spain, Dec. 2016. [Online]. Available: https://arxiv.org/abs/1701.00160

[23] N. Sebastian, B. Cseke, and R. Tomioka, "f-GAN: Training generative neural samplers using variational divergence minimization," *Proc. NIPS*, pp. 271–279, 2016.

[24] F. Cedric and J. Idier, "Algorithms for nonnegative matrix factorization with the β-divergence," *Neural computation*, vol. 23, no. 9, pp. 2421–2456, Aug. 2011.

[25] G. E. Henter, S. Ronanki, O. Watts, M. Wester, Z. Wu, and S. King, "Robust TTS duration modelling using DNNs," in *Proc. ICASSP*, Shanghai, China, Mar. 2016, pp. 5130–5134.

[26] R. Miyazaki, H. Saruwatari, T. Inoue, Y. Takahashi, K. Shikano, and K. Kondo, "Musical-noise-free speech enhancement based on optimized iterative spectral subtraction," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 7, pp. 2080–2094, Sep. 2012.

[27] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum likelihood estimation of spectral parameter trajectory," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2222–2235, 2007.

[28] O. Watts, Z. Wu, and S. King, "Sentence-level control vectors for deep neural network speech synthesis," in *Proc. INTERSPEECH*, Dresden, Germany, Sep. 2015.

[29] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilci, M. Sahidullah, and A. Sizov, "ASVspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge," in *Proc. INTERSPEECH*, Dresden, Germany, Sep. 2015, pp. 2037–2041.

[30] J. Lindberg and M. Blomberg, "Vulnerability in speaker verification - a study of technical impostor techniques," in *Proc. EUROSPEECH*, Budapest, Hungary, Mar. 1999, pp. 1211–1214.

[31] M. Abe, Y. Sagisaka, T. Umeda, and H. Kuwabara, "ATR technical report," no. TR-I-0166M, 1990.

[32] 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 *MAVEBA 2001*, Firentze, Italy, Sep. 2001, pp. 1–6.

[33] Y. Ohtani, T. Toda, H. Saruwatari, and K. Shikano, "Maximum likelihood voice conversion based on GMM with STRAIGHT mixed excitation," in *Proc. INTERSPEECH*, Pittsburgh, U.S.A., Sep. 2006, pp. 2266–2269.

[34] H. Kawahara, I. Masuda-Katsuse, and A. D. Cheveigne, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," *Speech Communication*, vol. 27, no. 3–4, pp. 187–207, 1999.

[35] S. Takamichi, K. Kobayashi, K. Tanaka, T. Toda, and S. Nakamura, "The NAIST text-to-speech system for the Blizzard Challenge 2015," in *Proc. Blizzard Challenge workshop*, Berlin, Germany, Sep. 2015.

[36] N. Hojo, Y. Ijima, and H. Mizuno, "An investigation of DNN-based speech synthesis using speaker codes," in *Proc. INTERSPEECH*, San Francisco, U.S.A., Sep. 2016, pp. 2278–2282.

[37] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. AISTATS*, Lauderdale, U.S.A., Apr. 2011, pp. 315–323.

[38] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Incorporation of mixed excitation model and postfilter into HMM-based text-to-speech synthesis," *IEICE Transactions on Information and Systems*, vol. J87-D-II, no. 8, pp. 1563–1571, 2004.

[39] T. Toda and K. Tokuda, "A speech parameter generation algorithm considering global variance for HMM-based speech synthesis," *IEICE Transactions on Information and Systems*, vol. E90-D, no. 5, pp. 816–824, 2007.

[40] S. Takamichi, T. Toda, A. W. Black, and S. Nakamura, "Parameter generation algorithm considering modulation spectrum for HMM-based speech synthesis," in *Proc. ICASSP*, Brisbane, Australia, Apr. 2015, pp. 4210–4214.