Denoising Recurrent Neural Network for Deep Bidirectional LSTM based Voice Conversion

Jie Wu[1], *Dongyan Huang*[2], *Lei Xie*[1], *Haizhou Li*[2,3]

[1]School of Computer Science, Northwestern Polytechnical University, Xi′an, China
[2]Institute for Infocomm Research, A*STAR, Singapore
[3] Department of Electrical and Computer Engineering, National University of Singapore

jiewu@nwpu-aslp.org, huang@i2r.a-star.edu.sg, lxie@nwpu.edu.cn, eleliha@nus.edu.sg

Abstract

The paper studies the post processing in deep bidirectional Long Short-Term Memory (DBLSTM) based voice conversion, where the statistical parameters are optimized to generate speech that exhibits similar properties to target speech. However, there always exists residual error between converted speech and target one. We reformulate the residual error problem as speech restoration, which aims to recover the target speech samples from the converted ones. Specifically, we propose a denoising recurrent neural network (DeRNN) by introducing regularization during training to shape the distribution of the converted data in latent space. We compare the proposed approach with global variance (GV), modulation spectrum (MS) and recurrent neural network (RNN) based postfilters, which serve a similar purpose. The subjective test results show that the proposed approach significantly outperforms these conventional approaches in terms of quality and similarity.

Index Terms: residual error, Gaussian noise, denoising, recurrent neural network, voice conversion

1. Introduction

Voice conversion (VC) [1] is a technique to modify one speaker's (source) voice to impersonate another speaker (target) while keeping its linguistic information unchanged. VC is useful in many tasks, such as personalized text-to-speech (TTS) synthesis [2], speech enhancement [3], speech-to-speech translation [4] and so on.

Recently, deep neural network (DNN) and deep bidirectional long short term memory (DBLSTM) were proposed as effective non-linear voice conversion models [5, 6, 7], which use several hidden layers in the conversion architecture to capture speech characteristics. Especially, the DBLSTM architecture can store long-range segmental information in its memory blocks and peephole connections to learn the contextual dependency of speech [8], which achieves superior performance to other competitive models [9], e.g., Gaussian mixture model (GMM) [10], dynamic kernel partial least square (DKPLS) [11], nonnegative matrix factorization (NMF) [12] etc. We note that neural network based approaches usually result in muffled speech due to *over-smoothness*, that is, the converted parameters are the averaging of the parameters of the trained model.

Post-processing has been a popular way to solve the over-smoothing issue, such as global variance (GV) [13, 14] and modulation spectrum (MS) [15, 16] based postfilters. These postfilters aim to adjust the variance of the converted parameters to match that of target parameters either in the mel-cepstra domain or in the modulation spectrum domain to enhance its dynamic property. However, these approaches are based on empirical findings of the difference between the converted and the target parameters. Recently, DNN-based postfilter [17, 18] has been proposed to learn the difference directly from data by mapping the converted parameters to the target parameters. Considering the high temporal correlation of speech, it is more appropriate to use recurrent neural network (RNN) to handle the time dependency between consecutive speech frames [19]. Hence RNN-based postfilter [20] has achieved better performance. Apparently, these postfilters aim to post-process the converted speech by compensating the residual error, i.e., compensating the empirical mismatch between the converted speech and the target speech.

In this paper, different from residual error compensation, we reformulate the residual error problem as a speech restoration task. Specifically, we consider the voice conversion process as a local corrupted process [21] [22], where converted parameters are considered as corrupted target parameters. We propose to use a denoising recurrent neural network (namely DeRNN) to restore the original speech (i.e., the target speech) from the corrupted speech. Our work is also motivated by recent advances of RNN in speech enhancement and robust speech recognition, which serves as a similar purpose [23, 24]. Subjective test results on DBLSTM-based voice conversion show that the proposed DeRNN approach significantly outperforms the conventional GV, MS and RNN postfilters in terms of both quality and similarity.

2. DBLSTM-based voice conversion

As we use DBLSTM as our voice conversion model, in this section, we first briefly discuss its inherent mismatch between the converted parameters and the target parameters as well as the ways to handle the mismatch.

The framework of a typical DBLSTM-based voice conversion approach [7] is illustrated in Fig. 1, which has a training stage and a conversion stage. At the training stage, we use WORLD [25] to extract spectral envelope from speech and then mel-cepstral coefficients (MCCs) [26] are extracted from the spectral envelope. Next, parallel MCCs of the source and target speech for training are aligned through dynamic time warping (DTW). Then, the aligned source and target MCCs are used as the input and output features to train the DBLSTM model by the back-propagation through time (BPTT) algorithm. A nonlinear mapping function is learned from source MCCs X to target MCCs Y, which can be formulated as:

$$\hat{Y} = F(X) \tag{1}$$

During the conversion process, given the MCCs X, a new source speech sequence, its corresponding converted MCCs $\hat{Y}$ are predicted by the learned DBLSTM mapping function $F(\cdot)$ frame by frame. Consequently, the WORLD is used as the

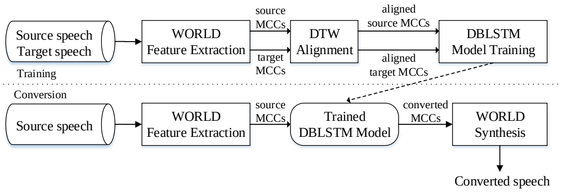


Figure 1: *The architecture of a typical DBLSTM-based voice conversion approach.*

vocoder to synthesize the converted speech from the predicted MCCs $\hat{Y}$.

In spite of its superior performance, there always exists an error between the converted parameters $\hat{Y}$ and target parameters Y according to Eq. (1). This error degrades both quality and similarity of the converted speech. To reduce such error, one can add extra hidden layers or adopt a tandem approach [27, 28] to build a tandem DBLSTM architecture as a refining model to predict the target parameters based on the converted speech. RNN-based postfilter [20] can be considered as an example of such refining model. Hence, with learning rate α, the weights w_{lm} (from unit l to unit m) can be updated in the error back-propagation process of the refining model as:

$$w_{lm} = w_{lm} - \alpha \frac{\partial E}{\partial w_{lm}} \tag{2}$$

$$E = (Y - \hat{Y})^T (Y - \hat{Y}) \tag{3}$$

where E is the squared error, Y, $\hat{Y}$ are the target and predicted parameters respectively. As Y is quite close to $\hat{Y}$, E is extremely small and thus may result in vanishing gradient problem that prevents the weights w_{lm} to converge to global minimum value.

In other words, we can consider $\hat{Y}$ as Y corrupted by additive noise, if $\hat{Y}$ and Y are close to each other. Therefore, the conversion process can be considered as a local corruption process where target parameters are corrupted by additive noise. It can be straightforward to reconstruct the target parameters from the converted ones.

3. DeRNN for DBLSTM based voice conversion

We now discuss formulating the residual error problem as speech restoration and employ a denoising recurrent neural network approach for speech restoration, which we call DeRNN.

3.1. Residual Error

We define residual error $\epsilon(\hat{Y}) = [\epsilon(1), \cdots, \epsilon(d), \cdots, \epsilon(D)]$ by subtracting target parameters $Y = [y(1), \cdots, y(d), \cdots, y(D)]$ from the converted ones $\hat{Y} = [\hat{y}(1), \cdots, \hat{y}(d), \cdots, \hat{y}(D)]$ as:

$$\epsilon(d) = \hat{y}(d) - y(d) \tag{4}$$

where $\epsilon(d)$ is the d-th dimension residual error between the d-th dimension converted parameters and the d-th dimension target parameters in latent space. D is the dimensionality.

If $\hat{y}(d)$ and $y(d)$ are close to each other, the residual error can be considered as Gaussian noise and formulated with a mean 0 and variance σ^2 as:

$$\epsilon(d) = \hat{y}(d) - y(d) \sim N(0, \sigma^2) \tag{5}$$

From Eq.(5), the converted parameters are reformulated as:

$$\hat{y}(d) = y(d) + \epsilon(d), \epsilon(d) \sim N(0, \sigma^2) \tag{6}$$

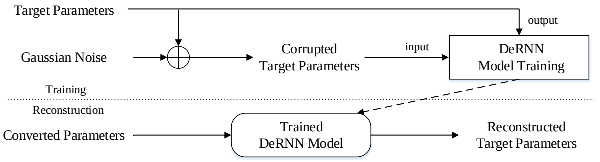


Figure 2: *Diagram of the DeRNN approach for reconstructing original target parameters from the converted ones.*

where $y(d) + \epsilon(d)$ is seen as the original target parameters corrupted by additive Gaussian noise. The converted parameters $\hat{y}(d)$ can be therefore considered as the corrupted target parameters. In this way, we reformulate the residual error problem as speech restoration such that original target parameters $y(d)$ can be reconstructed from the corrupted ones $\hat{y}(d)$.

We can consider the conversion process as the encoding process of auto-encoder [21]: a local corruption process, defined as $C(\hat{Y}|X)$ where X is source parameters and $\hat{Y}$, the corrupted Y is the auxiliary variable in latent space. The decoding process is to train the conditional distribution $p_\theta(Y|\hat{Y})$ on the data pairs, $\{(\hat{y}(d), y(d))\}_{d=1}^{D}$. We propose the following algorithm to learn $p_\theta(Y|\hat{Y})$.

Algorithm: The training algorithm for speech restoration requires a training set of original target examples Y and converted examples $\hat{Y}$ to train a conditional distribution $p_\theta(Y|\hat{Y})$.

```
Repeat
    • Original target examples $Y$
    • Corrupted target examples (converted examples) $\hat{Y} \sim C(\hat{Y}|X)$
    • Use ($\hat{Y}$,$Y$) as training examples to maximize the
      expected value of $p_\theta(Y|\hat{Y})$, e.g., by stochastic gradient
      descent approximation with respect to $\theta$.
until convergence of training
```

3.2. DeRNN

The denoising recurrent neural network (DeRNN) is used as the decoding process to train the $p_\theta(Y|\hat{Y})$ for speech restoration . The diagram of DeRNN is illustrated in Fig 2, which describes the training and reconstruction stages.

3.2.1. Training Stage

At the training stage, to shape the distribution of converted parameters during training, random Gaussian noise is used to simulate the distribution of residual errors. We add the random Gaussian noise to the original target parameters (Y) to simulate the corrupted target parameters ($\hat{Y}$). A nonlinear mapping function $f(\hat{Y}) \rightarrow Y$ is learned by DeRNN model from $\hat{Y}$ to Y that describes the relationship between corrupted target parameters and the desired target parameters..

Specifically, the inner structure of DeRNN is shown in Fig. 3, where the input makes use of contextual information for taking three frames of the corrupted target parameters.

As an example to illustrate, we show the distributions of the residual error from training data and its corresponding random Gaussian noise in Fig. 4, respectively.

3.2.2. Reconstruction Stage

At the reconstruction stage, the converted parameters are directly fed into the trained DeRNN model to reconstruct its correspond-

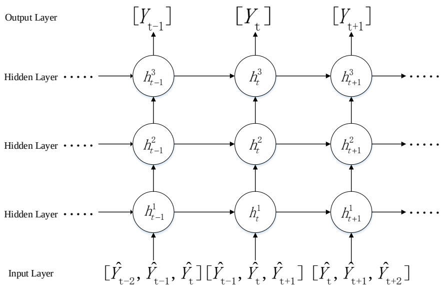
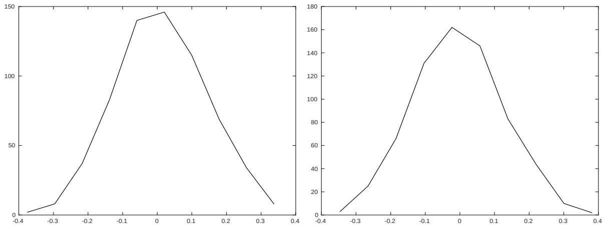


Figure 3: *Structure of the DeRNN model. A model with 3 hidden layers that takes 3 frames of corrupted target parameters and predicts original target parameters of the center frame.*



(a) *Residual error* (b) *Random Gaussian noise*

Figure 4: *Distributions of 11th dimension of the residual error and random Gaussian noise for the same sentence of training data in DBLSTM based voice conversion.*

ing original target parameters.

3.3. Spectra Feature Selection

DeRNN can be applied to any features. In this paper, we propose to employ DeRNN in the mel-cepstra domain for reconstructing original target MCCs from the converted MCCs.

3.3.1. Motivation

In signal processing, mel-cepstrum [26] is used to represent the short-term power spectrum of a signal based on nonlinear mel scale of frequency, which is derived as a nonlinear Fourier transform of the spectrum. Mel cepstral coefficients (MCCs) are coefficients of the mel-frequency spectrum, which can efficiently reflect the human auditory system's response.

3.3.2. Method

The residual errors for the training data are obtained by subtracting the aligned target MCCs from the aligned converted MCCs. Next, we produce random Gaussian noise with mean zero and statistical variances of these residual errors and then add it to the original target MCCs to form corrupted target MCCs. The corrupted and original target MCCs are used as the input and output features of the DeRNN respectively.

3.4. DeRNN vs GV, MS, and RNN postfilters

The DeRNN technique is similar to GV, MS, and RNN based postfilters in the sense that it tries to rectify the converted speech. However, it is different from them in many ways.

GV-based postfilter [13, 14] is proposed in response to the observation that the variance of converted parameters is smaller than that of the target ones. It forces the variance of converted mel-cepstra closer to that of the target ones by linear mapping. MS-based postfilter [15, 16] is proposed to enhance the dynamics of converted parameters in modulation spectrum domain by forcing the variance of converted modulation spectrum closer to that of the target ones. These postfilters are both based on empirical finding where the difference between variances tends to occur for most speakers. The RNN-based postfilter [20] is proposed to predict original parameters based on the converted parameters for residual error compensation. In our proposed approach, the conversion process is considered as a noisy channel, through which target parameters are corrupted by additive Gaussian noise. The DeRNN model is employed for recovering original target parameters from the converted ones regardless of specific values of the residual errors.

4. Experiments

4.1. Experimental Setup

In our experiments, we use CMU ARCTIC corpus [29] for intra-gender and inter-gender conversion. For DBLSTM-based voice conversion, in pre-training, 450 and 50 parallel sentences randomly-selected are used as training and validation data. 20 and 5 parallel sentences are used as training and validation data to re-train. In addition, 10 sentences are used as evaluation data during re-training. For DeRNN, 450 and 50 sentences from the pre-training are used as training and validation data, 10 sentences converted from the re-training are used as evaluation data. Speech signals are sampled of 16kHz and the size of window is 25ms with 5ms frame shift. We use WORLD [25] to extract spectral envelope, aperiodic component (AP) and $LogF_0$. 49-dim MCCs (except for energy dimension) extracted from the spectral envelope are converted by DBLSTM-based voice conversion system, $LogF_0$ is linearly converted and then we copy the AP of source speech to synthesize the converted speech.

The DBLSTM-based voice conversion model and DeRNN model are both trained by toolkit MERLIN [30]. In the former, there are three hidden layers in the network where each hidden layer is bidirectional LSTM. In each layer, the number of units is set as [49, 96, 128, 96, 49] for pre-training and re-training. While DeRNN has three hidden layers in the network where each hidden layer is RNN. In order to take advantage of context information, three frames of corrupted target MCCs are used as input features. The number of units in each layer is [147, 1024, 2048, 1024, 49] respectively. The DBLSTM and DeRNN models are both trained by BPTT algorithm and optimized by stochastic gradient descent approximation with a learning rate of 0.001 and momentum of 0.5.

We implement five systems for comparison:

- **NONE**: The typical DBLSTM-based voice conversion [7] without post-processing.
- **GV**: GV-based postfilter [14] is adopted.
- **MS**: MS-based postfilter [16] is adopted.
- **RNN**: RNN-based postfilter [20] is adopted.
- **DeRNN**: The proposed postfilter, in which DeRNN is used in the mel-cepstra domain.

4.2. Objective Evaluation

4.2.1. Mel-cepstral distortion (MCD)

Mel-cepstral distortion (MCD) [31] is used as objective measure of the spectral distance from converted to target speech, which is denoted as:

$$MCD[dB] = \frac{10}{ln10}\sqrt{2\sum_{d=1}^{D}(C_d^{target} - C_d^{converted})^2} \quad (7)$$

where C_d^{target} and $C_d^{converted}$ are the d-th coefficient of the target and converted MCCs, respectively. d is the dimension

Table 1: *The MCD of the aforementioned five different systems.*

Conversion	Female-Female	Female-Male	Male-Female	Male-Male
Speaker-pair	CLB-SLT	CLB-BDL	RMS-SLT	RMS-BDL
Source-Target	7.230	9.296	9.574	8.695
NONE	5.403	5.748	5.588	5.833
GV	5.688	5.979	5.641	6.013
MS	6.235	6.120	6.110	6.172
RNN	5.183	5.728	5.678	5.875
DeRNN	6.404	6.960	6.714	7.044

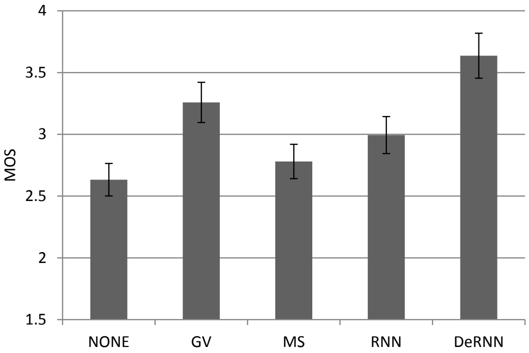


Figure 5: *Mean opinion scores (MOS) test results with* 95% *confidence intervals for speech quality.*

index and D is the dimensionality of MCCs (except for energy dimension). We expect a good postfilter to report a low MCD value.

The MCD scores of the above five systems for intra-gender and inter-gender voice conversion are summarized in Table 1. We note that the MCD score of DeRNN is the highest, which shows that the reconstructed target speech may still contain some noise. As presented in [32], each original speech (i.e., original target speech) has a noise masking threshold. If additive noise imposed on the original speech is below the masking threshold, it will be inaudible. We hope that the artifacts introduced by DeRNN are below such a threshold, thus are inaudible. This will be confirmed in the subjective listening tests. We understand that objective measures might not have direct correlation with human listening tests. Objective measure provides a practical way to tune parameters [33]. The following subjective listening results would support our explanations.

4.3. Subjective Evaluation

To evaluate the quality and similarity of the converted speech from these five systems, we conduct a subjective listening test for Female-Male voice conversion and 20 listeners are invited to evaluate 10 sentences in each system.

We carry out Mean Opinion Score (MOS) test for evaluating speech quality. In the MOS test, comparing with target speech, the grades of the converted speech are: 5 = excellent, 4 = good, 3 = fair, 2 = poor, and 1 = bad. Meanwhile, ABX preference test is adopted to evaluate speaker similarity of the converted speech among different systems. The results of MOS scores for speech quality are shown in Fig. 5 and the preference bars for speaker similarity are shown in Fig. 6.

4.3.1. DeRNN vs. NONE

Firstly, we would like to see the effectiveness of our proposed approach in improving the performance of DBLSTM-based voice conversion system. For speech quality, we note that the MOS score of DeRNN is significantly higher than that of NONE shown in Fig. 5. Meanwhile, according to the first bar in Fig. 6, it is obvious that DeRNN achieves much better performance than NONE in similarity. These results confirm the effectiveness of our proposed post-processing approach.

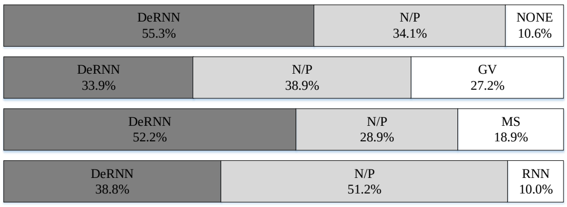


Figure 6: *ABX preference test results for speaker similarity. The p-values for the four bars are* $1.36 * 10^{-20}$, 0.171, $1.07 * 10^{-11}$, $2.18 * 10^{-10}$ *respectively.*

4.3.2. DeRNN vs. GV

We compare DeRNN with GV in speech quality. We note that DeRNN achieves a higher MOS score than that of GV method shown in Fig. 5. The performance of DeRNN is better than that of GV in similarity as well according to the second bar in Fig. 6. The results indicate that DeRNN outperforms GV-based postfilter in both quality and similarity of the converted speech.

4.3.3. DeRNN vs. MS

We study the performances of DeRNN and MS shown in Fig. 5 and the third bar in Fig. 6. The results show that DeRNN significantly outperforms MS in both speech quality and similarity.

4.3.4. DeRNN vs. RNN

Finally, DeRNN outperforms the RNN-based postfilter in both quality and similarity in the MOS scores and the similarity preference shown in Fig. 5 and the last bar in Fig. 6, respectively. Moreover, the RNN-based postfilter achieves a higher MOS score than NONE. It shows that RNN as a refining model is efficient to some extent while its weights do not guarantee convergence to the global minimum value.

5. Conclusions

In the paper, we define the residual error and treat the conversion process of DBLSTM based voice conversion as a local corrupted process, where the converted parameters are considered as the corrupted target parameters. The denoising RNN (DeRNN) model is proposed to be a post-processing filter for reconstructing original target parameters from the converted ones. Results show that our proposed approach can achieve superior performance than other conventional approaches in terms of quality and similarity of the converted speech. In the future, we will apply denoising adversarial auto-encoders [22] to the residual error problem for further improving the quality and similarity of the converted speech. Some samples for the subjective listening test are available via this link:`http://www.nwpu-aslp.org/attachments/INTERSPEECH2017-DeRNN-Demo.pptx`

6. Acknowledgements

This research is supported by Ministry of Education, Singapore AcRF Tier 1 NUS Start-up Grant FY2016 and the National Natural Science Foundation of China (Grant No. 61571363).

7. References

[1] S. H. Mohammadi and A. Kain, "An overview of voice conversion systems," *Speech Communication*, vol. 88, pp. 65–82, 2017.

[2] A. Kain and M. W. Macon, "Spectral voice conversion for text-to-speech synthesis," in *Acoustics, Speech and Signal Processing*, vol. 1. IEEE, 1998, pp. 285–288.

[3] D. Hironori, K. Nakamura, T. Tomoki, H. Saruwatari, and K. Shikano, "Esophageal speech enhancement based on statistical voice conversion with gaussian mixture models," *IEICE TRANSACTIONS on Information and Systems*, vol. 93, no. 9, pp. 2472–2482, 2010.

[4] N. Hattori, T. Toda, H. Kawai, H. Saruwatari, and K. Shikano, "Speaker-adaptive speech synthesis based on eigenvoice conversion and language-dependent prosodic conversion in speech-to-speech translation," in *Proc. INTERSPEECH*. ISCA, 2011, pp. 2769–2772.

[5] T. Nakashika, R. Takashima, T. Takiguchi, and Y. Ariki, "Voice conversion in high-order eigen space using deep belief nets," in *Proc. INTERSPEECH*. ISCA, 2013, pp. 369–372.

[6] L.-H. Chen, Z.-H. Ling, L.-J. Liu, and L.-R. Dai, "Voice conversion using deep neural networks with layer-wise generative training," *Trans. Audio, Speech & Language Processing*, vol. 22, no. 12, pp. 1859–1872, 2014.

[7] L. Sun, S. Kang, K. Li, and H. M. Meng, "Voice conversion using deep bidirectional long short-term memory based recurrent neural networks," in *Proc. ICASSP*. IEEE, 2015, pp. 4869–4873.

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] D. Huang, L. Xie, Y. S. W. Lee, J. Wu, H. Ming, X. Tian, S. Zhang, C. Ding, M. Li, Q. H. Nguyen *et al.*, "An automatic voice conversion evaluation strategy based on perceptual background noise distortion and speaker similarity," in *Proc. SSW*. ISCA, 2016, pp. 46–53.

[10] T. Toda, J. Lu, H. Saruwatari, and K. Shikano, "Straight-based voice conversion algorithm based on gaussian mixture model," in *Proc. INTERSPEECH*. ISCA, 2000, pp. 279–282.

[11] E. Helander, H. Silén, T. Virtanen, and M. Gabbouj, "Voice conversion using dynamic kernel partial least squares regression," *Trans. Audio, Speech & Language Processing*, vol. 20, pp. 806–817, 2012.

[12] Z. Wu, T. Virtanen, E. S. Chng, and H. Li, "Exemplar-based sparse representation with residual compensation for voice conversion," *Trans. Audio, Speech & Language Processing*, vol. 22, no. 10, pp. 1506–1521, 2014.

[13] T. Tomoki and K. Tokuda, "A speech parameter generation algorithm considering global variance for hmm-based speech synthesis," *IEICE TRANSACTIONS on Information and Systems*, vol. 90-D, no. 5, pp. 816–824, 2007.

[14] T. Toda, T. Muramatsu, and H. Banno, "Implementation of computationally efficient real-time voice conversion," in *Proc. INTERSPEECH*. ISCA, 2012, pp. 94–97.

[15] S. Takamichi, T. Toda, G. Neubig, S. Sakti, and S. Nakamura, "A postfilter to modify the modulation spectrum in hmm-based speech synthesis," in *Proc. ICASSP*. IEEE, 2014, pp. 290–294.

[16] S. Takamichi, T. Toda, A. W. Black, and S. Nakamura, "Modulation spectrum-based post-filter for gmm-based voice conversion," in *Proc. APSIPA*. IEEE, 2014, pp. 1–4.

[17] L.-H. Chen, T. Raitio, C. Valentini-Botinhao, J. Yamagishi, and Z.-H. Ling, "Dnn-based stochastic postfilter for hmm-based speech synthesis." in *Proc. INTERSPEECH*. ISCA, 2014, pp. 1954–1958.

[18] L.-H. Chen, T. Raitio, C. Valentini-Botinhao, Z.-H. Ling, and J. Yamagishi, "A deep generative architecture for postfiltering in statistical parametric speech synthesis," *Trans. Audio, Speech & Language Processing*, vol. 23, no. 11, pp. 2003–2014, 2015.

[19] Y. Fan, Y. Qian, F. Xie, and F. K. Soong, "Tts synthesis with bidirectional lstm based recurrent neural networks," in *Proc. INTERSPEECH*. ISCA, 2014, pp. 1964–1968.

[20] P. K. Muthukumar and A. W. Black, "Recurrent neural network postfilters for statistical parametric speech synthesis," *CoRR*, vol. abs/1601.07215, 2016.

[21] Y. Bengio, L. Yao, G. Alain, and P. Vincent, "Generalized denoising auto-encoders as generative models," in *In: Advances in Neural Information Processing Systems*, 2013, pp. 899–907.

[22] A. Creswell and A. A. Bharath, "Denoising adversarial autoencoders," *arXiv preprint arXiv:1703.01220*, 2017.

[23] A. L. Maas, Q. V. Le, T. M. O'Neil, O. Vinyals, P. Nguyen, and A. Y. Ng, "Recurrent neural networks for noise reduction in robust asr." in *Proc. INTERSPEECH*. ISCA, 2012, pp. 22–25.

[24] F. Weninger, H. Erdogan, S. Watanabe, E. Vincent, J. Le Roux, J. R. Hershey, and B. Schuller, "Speech enhancement with lstm recurrent neural networks and its application to noise-robust asr," in *LVA/ICA*. Springer, 2015, pp. 91–99.

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

[26] S. Imai, "Cepstral analysis synthesis on the mel frequency scale," in *Proc. ICASSP*, vol. 8. IEEE, 1983, pp. 93–96.

[27] H. Hermansky, D. P. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional hmm systems," in *Proc. ICASSP*, vol. 3. IEEE, 2000, pp. 1635–1638.

[28] D. P. Ellis, R. Singh, and S. Sivadas, "Tandem acoustic modeling in large-vocabulary recognition," in *Proc. ICASSP*, vol. 1. IEEE, 2001, pp. 517–520.

[29] J. Kominek and A. W. Black, "The cmu arctic speech databases," in *Proc. SSW*. ISCA, 2004, pp. 223–224.

[30] Z. Wu, O. Watts, and S. King, "Merlin: An open source neural network speech synthesis system," in *Proc. SSW*. ISCA, 2016, pp. 218–223.

[31] R. Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *Communications, Computers and Signal Processing*, vol. 1. IEEE, 1993, pp. 125–128.

[32] J. D. Johnston, "Transform coding of audio signals using perceptual noise criteria," *Journal on selected areas in communications*, vol. 6, no. 2, pp. 314–323, 1988.

[33] Z. Wu and S. King, "Improving trajectory modelling for dnn-based speech synthesis by using stacked bottleneck features and minimum generation error training," *Trans. Audio, Speech & Language Processing*, vol. 24, no. 7, pp. 1255–1265, 2016.