Speech Synthesis Based on Deep Neural Networks with Direct Modeling of Amplitude Spectra

Ranniery Maia and Rui Seara

***Abstract*— In recent state-of-the-art text-to-speech systems, usually a sequence of graphemes is directly mapped onto the speech waveform using deep neural networks. Despite reaching very high quality, these approaches tend to be computationally costly at synthesis time and its training implementation is usually not trivial. In this paper, a method which can be interpreted as a simplified version of these systems is proposed. Here, frame-based smoothed log spectra, fundamental frequency, and phase information are modeled at training time, while synthesis runs in a straightforward fashion. Experiments show that the proposed approach outperforms traditional ones using acoustic modeling of speech features.**

***Keywords*— Deep learning, deep neural networks, speech synthesis, text-to-speech (TTS) systems.**

I. INTRODUCTION

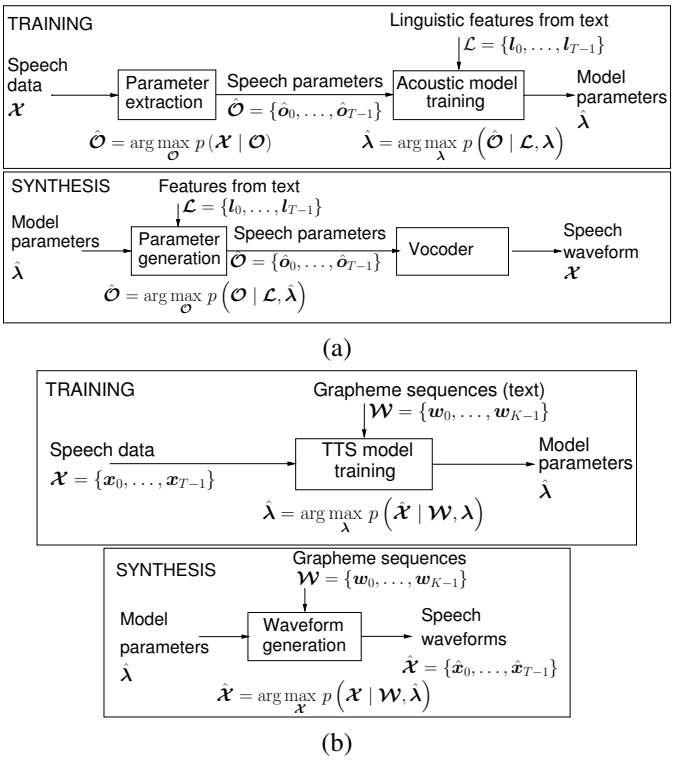
Statistical parametric speech synthesis (SPSS) [1] presents certain advantages when compared with the speech synthesis approaches which use concatenation of phonetic units [2]. Among them, we can highlight the capability of synthesizing speech with different voice styles through the manipulation of the model parameters. Besides, text-to-speech (TTS) systems based on SPSS can usually be built in an automatic way with a small amount of heuristic procedures.

Recently, important advances in SPSS have been achieved with application of deep learning [3]–[9]. The vast literature shows that deep learning improves quality when compared with state-of-the-art equivalent TTS systems based on hidden semi-Markov models (HSMM) by means of better representation of speech signal parameters. It also increases flexibility in terms of adaptation and creation of several voice styles. More recently, methods based on the concept of direct modeling of the speech waveform have been the trend, resulting in a framework known as *end-to-end speech synthesis* [10]–[15]. Among those techniques, *WaveNet* [10] represents a major breakthrough, since it has shown that deep convolutional networks can effectively model sequences that present an autoregressive structure, given past samples of the sequence and some *auxiliary parameters*. Because *WaveNet* is basically a general autoregressive network which can be used for any sort of data, several methods have used *WaveNet* as a vocoder where linguistic features are fed to the network as auxiliary features. Another breakthrough in the speech synthesis field has occurred with the publication of *Tacotron* [11], which is an end-to-end approach that maps character embeddings (graphemes or phonemes) onto Mel log speech spectra. *Tacotron* is basically a sequence-to-sequence architecture with modified encoder and decoder. More recently, an improved version of *Tacotron* [15] predicts speech spectrograms that are fed to a *WavNet* module, finally turning into an actual end-to-end synthesizer, with quality that is indistinguishable from that of a human being. The main advantage of the use of end-to-end systems regards the level of naturalness of synthesized speech, which is higher than the current state-of-the-art approach based on unit selection and concatenation [2]. In addition, greater flexibility in terms of synthesizing different speech styles and expressions. Such a flexibility can be achieved through the use of style embeddings that are extracted by neural networks [16]. Therefore, despite apparently solving the TTS problem, these end-to-end systems have in common one aspect: they are computationally complex and one usually takes quite a lot of empirical effort to reach the same results that are reported in the papers. Besides, even the best quality TTS systems show issues related to the time consumed to synthesize a single sentence.

The idea proposed in this paper is located in a *half-way* between traditional TTS methods (mapping between linguistic features and acoustic parameters) and recent approaches that outperform waveform concatenation systems, i.e., end-to-end synthesis with direct speech modeling. The proposed idea here has the advantage of being simple while maintaining a high level of synthesized speech quality. In fact, similar techniques have been proposed with the same purpose, see [17], [18]. In [17], speech amplitude spectra are directly modeled through several layers of feed-forward neural networks. At the input, aside from the usual linguistic features, the authors have also used logarithm of fundamental frequencies, $\ln(F_0)$, and voicing decision (VUV). In [18], the authors replace the acoustic features by amplitude and phase spectra on a warped domain, together with $\ln(F_0)$ and VUV. The disadvantage of [17] is that an external prosody model is needed in order to produce the network input at synthesis time. Our method is similar to [18]. However, here we use the frame-based modeling period, and to recover phase information at synthesis time we use the anti-causal portion of the complex cepstrum as phase feature [19].

This paper is organized as follows. Section II presents a brief historical background of TTS systems based on deep neural networks. Section III describes our TTS method which is based on the direct modeling of log amplitude spectra with phase recovery using anti-causal cepstrum. Experiments are presented in Section IV and the conclusions are in Section V.

R. Maia and R. Seara are with LINSE - Circuits and Signal Processing Laboratory, Department of Electrical and Electronics Engineering, Federal University of Santa Catarina (UFSC), Florianopolis-SC, Brazil, E-mails: rmaia@linse.ufsc.br, seara@linse.ufsc.br. This work was partially supported by the Brazilian National Council for Scientific and Technological Development (CNPq).



(a)

(b)

Fig. 1. TTS approaches. (a) Conventional one based on the mapping between linguistic and acoustic features. (b) End-to-end synthesis method with direct waveform modeling.

II. Speech Synthesis based on Deep Learning

The paper by Ling et al. [20] provides a very good historical background on the application of deep learning to TTS. In its simplest form, deep neural network (DNN) based TTS (DNN-TTS) systems can be implemented by replacing the typical decision trees from the HSMMs by feed-forward deep neural networks, as proposed in [6]. Nowadays, one can classify DNN-TTS methods into two categories. Conventional approaches that rely on the one-to-one mapping between linguistic features from text and acoustic features from speech; and the end-to-end approaches, which are based on the inference of speech waveforms from input text. Fig. 1 presents an overview on how both techniques work.

A. End-to-end Methods

Recent DNN-TTS approaches are based on the training of layers of carefully elaborated neural networks, which take as input phoneme or grapheme sequences (no need of syllable, word, phrase information anymore) and as output the speech signal, see [11], [14], [15]. These families of synthesizers achieve the higher level of naturalness. For instance, *Tacotron* [15] produces synthetic speech which is indistinguishable to natural, reaching up the same level (or even better than) unit concatenation-based systems.

In a probabilistic sense, training a TTS system can be viewed as the problem of finding, in a maximum likelihood sense, the optimal set of model parameters $\hat{\boldsymbol{\lambda}}$, so as

$$\hat{\boldsymbol{\lambda}} = \arg\max_{\boldsymbol{\lambda}} \; p\left(\boldsymbol{\mathcal{X}} \mid \mathcal{W}, \boldsymbol{\lambda}\right) p\left(\boldsymbol{\lambda}\right) \tag{1}$$

where $\boldsymbol{\mathcal{X}} = \{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{T-1}\}$ is a sequence of segments[1] of the speech waveform, with $\boldsymbol{x}_t = \begin{bmatrix} x_t(0) & \cdots & x_t(N_t-1) \end{bmatrix}^\top$ being the tth frame or segment of speech, N_t the corresponding number of samples and T the number of segments. The set of vectors $\mathcal{W} = \{\boldsymbol{w}_0, \ldots, \boldsymbol{w}_{K-1}\}$ is a sequence of graphemes, with $\boldsymbol{w}_k = \begin{bmatrix} w_k(0) & \cdots & w_k(M_k-1) \end{bmatrix}^\top$ denoting the kth grapheme sequence, M_k the corresponding number of graphemes and K the number of sequences. Operator $[\cdot]^\top$ indicates matrix transposition. At synthesis time, graphemes are fed to the trained neural network and the final speech waveform is obtained. In a probabilistic view this is written as

$$\hat{\boldsymbol{\mathcal{X}}} = \arg\max_{\boldsymbol{\mathcal{X}}} \; p\left(\boldsymbol{\mathcal{X}} \mid \mathcal{W}, \hat{\boldsymbol{\lambda}}\right). \tag{2}$$

Prior to the recent use of neural networks in TTS, some authors tried to implement the end-to-end concept, sometimes regarded as *waveform models* [21], [22].

B. Conventional Methods

Conventional TTS systems rely on the mapping between linguistic features and acoustic parameters, in which a DNN represents the probability of the acoustic features $\mathcal{O}$ given the input linguistic parameters $\mathcal{L}$, i.e.,

$$\hat{\boldsymbol{\lambda}} = \arg\max_{\boldsymbol{\lambda}} \; p\left(\mathcal{O} \mid \mathcal{L}, \boldsymbol{\lambda}\right) p\left(\boldsymbol{\lambda}\right). \tag{3}$$

The set of vectors $\mathcal{O} = \{\boldsymbol{o}_0, \ldots, \boldsymbol{o}_{T-1}\}$ usually contain speech acoustic parameters concatenated with their corresponding dynamic features. Thus,

$$\boldsymbol{o}_t = \begin{bmatrix} \boldsymbol{y}_t^\top & \Delta^{(1)}\boldsymbol{y}_t^\top & \cdots & \Delta^{(D)}\boldsymbol{y}_t^\top \end{bmatrix}^\top \tag{4}$$

where $\boldsymbol{y}_t$ is a vector containing parameters that can be used to reconstruct the speech signal using parametric models [23] and D is the dynamic feature order. The linguistic feature vectors $\mathcal{L} = \{\boldsymbol{l}_0, \ldots, \boldsymbol{l}_{T-1}\}$, on the other hand, contain linguistic features generated from the text database, in which each input vector $\boldsymbol{l}_t$ is composed of three sub-vectors, i.e.,

$$\boldsymbol{l}_t = \begin{bmatrix} \boldsymbol{l}_t^{(b)\top} & \boldsymbol{l}_t^{(n)\top} & \boldsymbol{l}_t^{(d)\top} \end{bmatrix}^\top \tag{5}$$

with $\boldsymbol{l}_t^{(b)}$, $\boldsymbol{l}_t^{(n)}$, and $\boldsymbol{l}_t^{(d)}$ representing, respectively, binary, numeric, and duration features. Note that in this approach there is a one-to-one vector matching, in contrast to end-to-end method, in which the number of grapheme sequences is different from the number of speech segments or samples.

During the synthesis, a sequence of linguistic features is generated from the text to be synthesized $\mathcal{L} = \{\boldsymbol{l}_0, \ldots, \boldsymbol{l}_{T-1}\}$, where T is the number of frames of the sentence to be synthesized. Variable $\mathcal{L}$ is then fed to the DNN, so that in the output it is produced another sequence of acoustic parameters, i.e.,

$$\hat{\mathcal{O}} = \arg\max_{\mathcal{O}} \; p\left(\mathcal{O} \mid \mathcal{L}, \hat{\boldsymbol{\lambda}}\right). \tag{6}$$

Because $\hat{\mathcal{O}}$ is assumed to be a sequence of stochastic means, the final acoustic feature sequence is obtained by applying

[1] Another way to see this formulation is to assume a vector of speech samples $\boldsymbol{x}^\top = \begin{bmatrix} x(0) & \cdots & x(N-1) \end{bmatrix}$, where in this case N is the number of samples.

one of the trajectory smoothing algorithms given in [24] on $\hat{\boldsymbol{\mathcal{O}}}$, taking into consideration the global variance, resulting in the final acoustic feature sequence $\hat{\boldsymbol{\mathcal{Y}}} = \{\hat{\boldsymbol{y}}_0, \ldots, \hat{\boldsymbol{y}}_{T-1}\}$. Lastly, the speech waveform is produced from $\hat{\boldsymbol{\mathcal{Y}}}$ assuming a parametric model of speech synthesis [23].

III. Proposed TTS Method

Our method is basically an improved version of the conventional TTS systems [6] outlined in Section II-B. Instead of using the acoustic features that are generally used in the conventional approach, we model the log Mel smoothed spectral envelope of speech, $\ln(F_0)$, and voicing decision. In addition, aiming to recover phase information, we add the anti-causal portion of the complex cepstrum as a parameter in the observation vector. Therefore, each vector $\boldsymbol{o}_t$ becomes

$$\boldsymbol{o}_t = \begin{bmatrix} \boldsymbol{S}_t & \mu_t & \ln(F_{0,t}) & \cdots & \Delta^{(D)} \ln(F_{0,t}) & \boldsymbol{\phi}_t \end{bmatrix}^\top \quad (7)$$

where μ_t is the voicing decision ($\mu_t = 1$ for voiced and $\mu_t = 0$ for unvoiced) for the tth frame, and

$$\boldsymbol{S}_t = \begin{bmatrix} \ln\left|H_t\left(e^{\jmath\tilde{\omega}_0}\right)\right| & \cdots & \ln\left|H_t\left(e^{\jmath\tilde{\omega}_P}\right)\right| \end{bmatrix}^\top \quad (8)$$

and

$$\boldsymbol{\phi}_t = \begin{bmatrix} \phi_t(1) & \cdots & \phi_t(C) \end{bmatrix}^\top \quad (9)$$

are, respectively, vectors containing the Mel log smoothed spectral envelope of speech with $\{\tilde{\omega}_0, \ldots, \tilde{\omega}_P\}$ being P warped scale angular frequencies, and phase features. The procedures of speech analysis and synthesis are discussed with more details in the following. Regarding the parts of linguistic feature extraction and statistical modeling, these procedures are carried out in the same way as in conventional TTS systems [6].

A. Speech Analysis

Training starts by extracting $\boldsymbol{\mathcal{L}}$ and $\boldsymbol{\mathcal{O}}$ from the database. Linguistic features $\boldsymbol{\mathcal{L}} = \{\boldsymbol{l}_0, \ldots, l_{T-1}\}$ are extracted from text and generated at the frame level. The elements that compose $\boldsymbol{\mathcal{O}}$ can be derived by the following procedures: 1) glottal close instant (GCI) detection [25]; and 2) complex cepstrum analysis [19]. Through the estimation of the GCI, $\{p_0, \ldots, p_{Z-1}\}$, where Z is the number of GCI, detection of the fundamental frequencies, $\{F_{0,0}, \ldots, F_{0,T-1}\}$, and voicing decisions, $\{\mu_0, \ldots, \mu_{T-1}\}$, can be obtained. Next, the frequency response of speech $s(n)$ at each instant p_z is determined by making

$$H_z\left(e^{\jmath\omega}\right) = \sum_{n=p_{z-1}}^{p_{z+1}} s(n) j\left(n - p_{z-1}\right) e^{-\jmath\omega n} \quad (10)$$

where $j(n)$ is an appropriate window to select $s(n)$ between p_{z-1} and p_{z+1}. Finally, the complex cepstrum is calculated by using

$$\hat{h}_z(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left\{\ln\left|H_z\left(e^{\jmath\omega}\right)\right| + \jmath\theta_z(\omega)\right\} e^{\jmath\omega n} d\omega \quad (11)$$

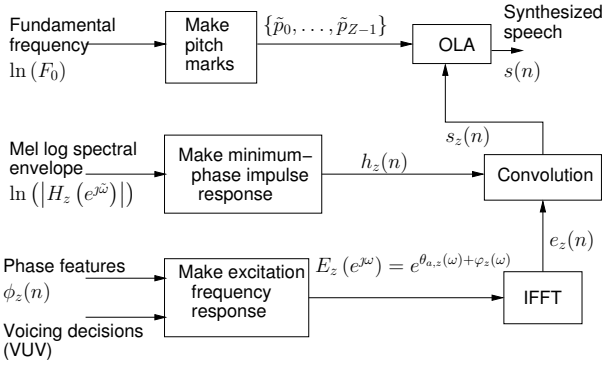


Fig. 2. Waveform generation from features derived from the DNNs.

where $|H_z(e^{\jmath\omega})|$ and $\theta_z(\omega)$ are, respectively, the amplitude and continuous phase responses of $s(n)$ at the instant p_z. The phase features at p_z are given by

$$\phi_z(n) = \hat{h}_z(-n-1), \quad 0 \le n < C. \quad (12)$$

Note that even though $\ln|H_z(e^{\jmath\omega})|$ and $\phi_z(n)$ are extracted pitch-synchronously, input-output mapping in the DNN during the training can be made in a periodic frame-based fashion[2]. To this end, pitch-synchronous parameters extracted at the instants $\{p_0, \ldots, p_{Z-1}\}$ can be transformed into frame-based ones by making

$$\boldsymbol{S}_t = \{\boldsymbol{S}_z \mid p_z \le tJ < p_{z+1}\} \quad (13)$$

and

$$\boldsymbol{\phi}_t = \{\boldsymbol{\phi}_z \mid p_z \le tJ < p_{z+1}\} \quad (14)$$

for $t = 0, \ldots, T-1$, where J is the frame duration in number of samples. Note that (14) simply repeats vectors from previous pitch to the next GCI.

B. Speech synthesis

Once $\hat{\boldsymbol{\mathcal{O}}}$ is estimated from the trained neural network, synthesis is performed as shown in Fig. 2. Initially, the estimated fundamental frequencies are used to generate the synthetic pitch marks $\{\tilde{p}_0, \ldots, \tilde{p}_{Z-1}\}$, and all parameters are sampled at those time instances. Then, for each position $\tilde{p}_z$, a synthetic speech segment $s_z(n)$ is given by the convolution of a minimum-phase impulse response $h_z(n)$ and an excitation signal $e_z(n)$, both at instant p_z, where

$$h_z(n) = \begin{cases} \exp\left(\hat{h}_z(0)\right) & n = 0 \\ \sum_{k=1}^{C} \frac{k}{n} \hat{h}_z(k) h_z(n-k) & n \ge 1 \end{cases} \quad (15)$$

and

$$\hat{h}_z(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \ln\left(\left|H_z\left(e^{\jmath\tilde{\omega}}\right)\right|\right) e^{\jmath\tilde{\omega}} d\omega, \;\; 0 \le n \le C \quad (16)$$

with $\ln\left(\left|H_z\left(e^{\jmath\tilde{\omega}}\right)\right|\right)$ being a generated Mel log spectral envelope. On the other hand, the excitation signal in the frequency domain $E_z(e^{\jmath\omega})$ is all-pass with phase response given by $\theta_z(\omega) = \theta_{a,z}(\omega) + \varphi_z(\omega)$. Component $\varphi_z(\omega)$ is zero when

[2]We use frame-based mapping because we utilize alignments provided by HSMM to produce the linguistic feature vectors $\boldsymbol{\mathcal{L}} = \{\boldsymbol{l}_0, \ldots, \boldsymbol{l}_{T-1}\}$.

the voicing decision is one and random when it is zero, while the residual phase $\theta_{a,z}(\omega)$ is calculated from generated phase features as

$$\theta_{a,z}(\omega) = -2\sum_{n=1}^{C_a} \phi_z(n)\sin(\omega n) \qquad (17)$$

where C_a is the number of phase features.

IV. Experiments

A. Experimental conditions

In order to test the proposed system, the database *Constituicao 1.0*, supplied by the group *FalaBrasil* [26] was used. This database consists of nine hours of audio sampled at 22.05 kHz recorded in a controlled environment by a male speaker. Text prompts are also provided alongside the corresponding audio. From the total amount of data, five hours have been used in our experiments. The process of database selection is similar to that shown in [27].

The phonetic labeling of the database has been obtained through a neural grapheme-phone conversion mechanism. Besides the G2P part, the remaining linguistic features (syllable, word, phrase, among others) have been derived from the Festival Speech Synthesis system [28]. Pitch marking has been carried out by SWIPE [29]. Pitch synchronous amplitude spectra and phase features have been extracted from the speech material using the method described in Section III, with $P = 256$ (Mel warped scale) and $C = 19$.

The neural network used to train the TTS system had two layers of 1024 feed-forward units, two layers of 512 LSTM units, and one output linear layer. All activation functions are hyperbolic tangent. The linguistic features are normalized between zero and one, while the outputs are normalized for mean zero and variance one.

For comparison purposes, a baseline system is trained under the same conditions, with the sole difference being the features that populated the sequence $\boldsymbol{\mathcal{O}}$, which consisted of 45 Mel cepstral coefficients derived from pitch-synchronous spectra, voicing decision, and fundamental frequency, with the corresponding delta and delta-delta features aside from the voicing decision. The generation part also takes into account the speech parameter generation algorithm, which is typically used in conventional TTS systems [24].

B. Objective analysis

The performance of the proposed and baseline systems have been assessed by using the following metrics: log spectral distortion (LSD), root mean squared of F_0 (RMS_{F_0}) in voiced regions (difference of fundamental frequencies), and percentage of frames with wrong voicing decisions.

TABLE I
Results of the objective evaluation

	Conventional	Proposed
LSD (dB)	7.51	**7.42**
RMS_{F_0} (Hz)	**20.11**	20.64
PCVUV (%)	**96.61**	96.17

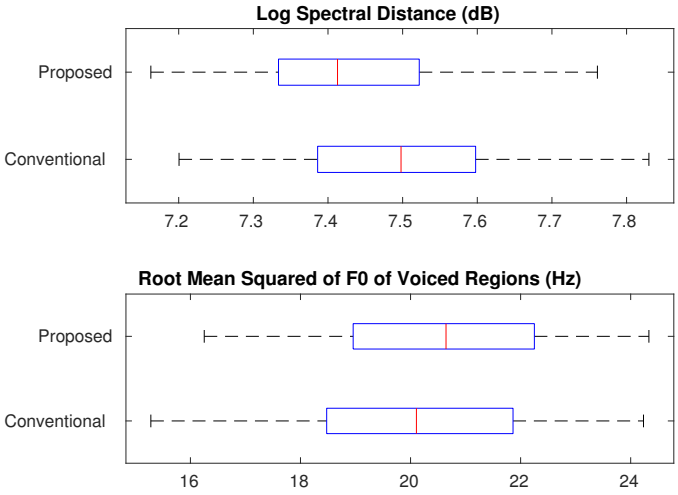


Fig. 3. Box-plot analysis of the samples used to calculate the medians shown in Table I. Top: log spectral distance. Bottom: root mean squared of fundamental frequency of voiced regions.

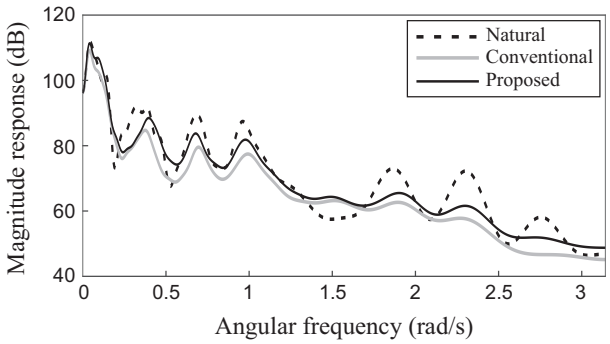


Fig. 4. Short-time speech spectral envelope from natural, and speech synthesized by conventional and proposed systems.

Twenty sentences have been used in the test, corresponding to more or less ten minutes of recorded speech taken from the same speaker. The final LSD is taken as the median of LSD of all frames, while the RMS_{F_0} is taken as the median of all twenty sentences. The overall percentage of correct voicing decisions (PCVUV) is taken as percentage of voicing decisions from all frames. Table I shows the objective results for the test set. It can be noticed that although the conventional method achieves better results in terms of RMS_{F_0} and PCVUV, these differences are mostly insignificant in terms of quality. On the other hand, a 0.1-dB difference in terms of LSD can have a significant impact on the quality of the synthetic speech [23]. Fig. 3 shows a box-plot analysis of the samples that are used to calculate the medians presented in Table I, where one can notice the tendency of lower LSD in our proposed approach. Fig. 4 shows short-term amplitude responses of natural and synthesized speech for both conventional and proposed TTS systems. It can be noticed that in this example the proposed system produces an spectral envelope with peaks that are closer to the ones present in the natural spectrum. That fact indicates that synthesized speech may sound less *muffled* when compared with the one generated by the conventional system.

V. CONCLUSIONS

In this paper, we propose a TTS system in which Mel log spectra is directly modeled as output parameter, together with fundamental frequency and voicing decision. In addition, to reconstruct phase information the anti-causal part of the complex cepstrum is also modeled by the DNNs. The proposed approach aims to synthesize speech with higher quality, close to recent state-of-the-art systems. It also intends to keeping the computational complexity at the same level as conventional DNN-TTS systems, which are based on the modeling of acoustic parameters and use a speech model to synthesize the final waveform. Experiments showed that the proposed approach outperforms the conventional approaches in terms of log spectral distortion. Continuation of this project includes switching from a one-to-one mapping between linguistic features and amplitude spectra to a sequence-to-sequence mapping. By doing that, we shall give another step towards the implementation of end-to-end systems.

REFERENCES

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

[2] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, (Atlanta, USA), pp. 373–376, May 1996.

[3] S. Kang, X. Qian, and H. Meng, "Multi-distribution deep belief network for speech synthesis," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, (Vancouver, Canada), pp. 8012–8016, 2013.

[4] Z.-H. Ling, L. Deng, and D. Yu, "Modeling spectral envelopes using restricted Boltzmann machines for statistical parametric speech synthesis," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, (Vancouver, Canada), pp. 7825–7829, May 2013.

[5] R. Fernandez, A. Rendel, B. Ramabhadran, and R. Hoory, "F0 contour prediction with a deep belief network-gaussian process hybrid model," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, (Vancouver, Canada), pp. 6885–6889, 2013.

[6] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, (Vancouver, Canada), pp. 7962–7966, May 2013.

[7] Y. Fan, Y. Qian, F. Xie, and F. K. Soong, "TTS synthesis with bidirectional LSTM based recurrent neural networks," in *Proc. Annual Conf. Int. Speech Communication Association (INTERSPEECH)*, (Singapore), pp. 1964–1968, Sept. 2014.

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

[9] Z. Wu, C. V. Botinhao, O. Watts, and S. King, "Deep neural network employing multi-tasked learning and stacked bottleneck features for speech synthesis," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, (Brisbane, Australia), pp. 4460–4464, Apr. 2015.

[10] 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," 2016. [Online]. Available: `https://arxiv.org/abs/1609.03499`.

[11] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *Proc. Annual Conf. Int. Speech Communication Association (INTERSPEECH)*, (Stockholm, Sweden), pp. 4006–4010, Aug. 2017.

[12] S. Arik, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, A. Ng, J. Raiman, S. Sengupta, and M. Shoeybi, "Deep voice: Real-time neural text-to-speech," in *Proceedings of the 34th International Conference on Machine Learning*, (Sydney, Australia), pp. 195–204, Aug. 2017.

[13] A. G., S. Arik, G. Diamos, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, "Deep voice 2: Multi-speaker neural text-to-speech," in *Advances in Neural Information Processing Systems 30*, pp. 2962–2970, 2017.

[14] W. Ping, K. Peng, A. Gibiansky, S. Arik, A. Kannan, and S. Narang, "Deep voice 3: Scaling text-to-speech with convolutional sequence learning," in *Proceedings of the Sixth International Conference on Learning Representations (ICLR)*, (Vancouver, Canada), pp. 1–16, Feb. 2018.

[15] J. Shen, R. Pang, R. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, R. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," in *Proc. Annual Conf. Int. Speech Communication Association (INTERSPEECH)*, (Calgary, Canada), Apr. 2018.

[16] Y. Wang, D. Stanton, Y. Zhang, R. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, F. Ren, Y. Jia, and R.-A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," 2018. [Online]. Available: `https://arxiv.org/abs/1803.09017`.

[17] S. Takaki, H. Kameoka, and J. Yamagish, "Direct modeling of frequency spectra and waveform generation based on phase recovery for dnn-based speech synthesis," in *Proc. Annual Conf. Int. Speech Communication Association (INTERSPEECH)*, (Stockholm, Sweden), pp. 1128–1132, Aug. 2017.

[18] F. Espic, C. Valentini-Botinhao, and S. King, "Direct modelling of magnitude and phase spectra for statistical parametric speech synthesis," in *Proc. Annual Conf. Int. Speech Communication Association (INTERSPEECH)*, (Stockholm, Sweden), pp. 1383–1387, Aug. 2017.

[19] R. Maia, M. Akamine, and M. Gales, "Complex cepstrum for statistical parametric speech synthesis," *Speech Communication*, vol. 5, pp. 606–618, June 2013.

[20] Z.-H. Ling, S. Kang, H. Zen, A. W. Senior, M. Schuster, X. 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.*, vol. 32, pp. 35–52, May 2015.

[21] R. Maia, H. Zen, and M. Gales, "Statistical parametric speech synthesis with joint estimation of acoustic and excitation model parameters," in *The Seventh ISCA Tutorial and Research Workshop on Speech Synthesis*, (Kyoto, Japan), pp. 88–93, Sept. 2010.

[22] K. Nakamura, K. Hashimoto, Y. Nankaku, and K. Tokuda, "Integration of spectral feature extraction and modeling for hmm-based speech synthesis," *IEICE Transactions on Information and Systems*, vol. 97-D, no. 6, pp. 1438–1448, 2014.

[23] J. R. Deller, Jr., J. H. L. Hansen, and J. G. Proaks, *Discrete-Time Processing of Speech Signals*. New York, NY, USA: IEEE Press Classic Reissue, 2000.

[24] K. Tokuda, T. Kobayashi, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, (Istanbul, Turkey), pp. 1315–1318, June 2000.

[25] T. Drugman, M. Thomas, J. Gudnason, P. Naylor, and T. Dutoit, "Detection of glottal closure instants from speech signals: a quantitative review," *IEEE Trans. Audio, Speech, Language Process.*, vol. 3, pp. 994–1006, Mar. 2012.

[26] "FalaBrasil: reconhecimento de voz para o português brasileiro." [Online]. Available: `http://www.laps.ufpa.br/falabrasil/`. Accessed Mar. 2017.

[27] R. Maia, J. Ni, S. Sakai, T. Toda, K. Tokuda, T. Shimizu, and S. Nakamura, "The NICT/ATR speech synthesis system for the Blizzard Challenge 2008." [Online]. Available: `http://festvox.org/blizzard/blizzard2008.html`. Accessed Apr. 2017.

[28] "The Festival Speech Synthesis System." [Online]. Available: `http://www.cstr.ed.ac.uk/projects/festival`. Accessed Apr. 2018.

[29] A. Camacho, *SWIPE: A Sawtooth Waveform Inspired Pitch Estimator for Speech and Music*. PhD thesis, University of Florida, 2007.