The CSTR entry to the 2018 Blizzard Challenge

Felipe Espic, Avashna Govender, Manuel Sam Ribeiro, Cassia Valentini-Botinhao, Oliver Watts

The Centre for Speech Technology Research (CSTR), University of Edinburgh, UK

owatts@inf.ed.ac.uk

Abstract

Similar to 2016 and 2017 Blizzard Challenge, the task for this year is to train on expressively-read children's story-books, and to synthesise speech in the same domain. This give us an opportunity to investigate the effectiveness of several techniques we have developed when applied to expressive and prosodically-varied audiobook data. This paper describes the text-to-speech system entered by The Centre for Speech Technology Research into the 2018 Blizzard Challenge. The system is a hybrid synthesis system where a halfphone unit selection synthesiser is driven by the output of a neural network based acoustic and duration model. We adopt the same neural network based models used in our last year entry with a different unit selection component. We discuss the performance of our system by reporting the results from formal listening tests provided by the challenge.

Index Terms: Merlin, hybrid speech synthesis, unit selection, deep neural networks.

1. Introduction

Our entry to this year's Blizzard Challenge closely follows the form of the hybrid systems we submitted for the previous two years' challenges [1, 2]. A major difference in implementation is that entirely different code was used for the unit selection component of the system: the codebase used to implement the systems described in [3, 4] was integrated into our Blizzard entry.

Hybrid synthesis systems compute a unit selection target cost [5, 6, 7, for example] using acoustic properties predicted for the speech to be produced. The idea is to combine the benefits of the stability of statistical parametric speech synthesis (SPSS) systems with the high quality waveforms of unit selection, unaffected by the degradations introduced by vocoding [8, 9]). Experiments presented in [6, 7] establish that improving the underlying SPSS of a hybrid synthesiser results in improvements to the concatenated output speech. The statistical models used in our entry this year are identical to those used in our entry to the 2017 challenge, and thus benefit from various incremental improvements made to them over the past years. In previous years, we incorporated various improvements to the underlying SPSS model compared to the system presented in [7]: the decision tree duration model was replaced with a bi-directional long short-term memory (LSTM) recurrent neural network, and the feed-forward DNN acoustic model was replaced with an LSTM network. Last year, the acoustic model's predictions of fundamental frequency were improved by the addition of supra-segmental features based on acoustic counts (see Section 2.3.3). The join-smoothing we employ (briefly described again in Section 2.6) is also inherited from last year's system. The neural networks used in this entry were trained using our open-source Merlin speech synthesis toolkit [10].

2. System Description

2.1. Data

Identical to the 2017 Challenge, the database – provided to the Challenge by Usborne Publishing Ltd. – consists of the speech and text of 56 children's audiobooks spoken by a British female speaker. Just as last year, we made use of a segmentation of the audiobooks carried out by two other Challenge participants[1][2] and kindly made available to other participants. The total duration of the audio is approximately 6 hours after segmentation. Three audiobooks from the given corpus were held out to act as an internal development set to gauge system performance before generating the final test data. The held-out data consists of three full short stories: *Goldilocks and the Three Bears*, *The Boy Who Cried Wolf* and *The Enormous Turnip*, having a total combined duration of approximately 10 minutes.

2.1.1. Sentence selection

For sentence selection, we have followed the same approach as for the 2016 and 2017 challenges. For clarity, we repeat the procedure followed in our previous entries [1, 2].

Harnessing the variety of speaking styles present in expressively-read audiobooks might enable us to produce less robotic-sounding TTS systems. However, initial experiments showed that the extreme variation in parts of the training data for the Challenge resulting in poor unit selection. We therefore filtered the data using the active learning approach described in [11]: 198 utterance-level acoustic features are extracted, and 15 sentences initially labelled as *keep* or *too expressive* by an expert listener. Uncertainty sampling [12] using an ensemble of decision trees was then used to select a further informative sample to be hand-labelled; this process continued for 20 minutes (real time). A classifier built on the entire set of hand-labelled data was then used to determine the subset of available sentences to be used for training. 20% of the training sentences were discarded in this way; informal comparison suggested this resulted in more stable synthesis with fewer unwarranted prosodic excursions.

2.2. Text processing

The text processing part of our system is identical to that used in the 2017 challenge [2], but we repeat a description of it here for completeness. We used Festival's English front-end with the British received pronunciation version of the Combilex lexicon [13]. 163 items were added to cover words appearing in the training data but otherwise absent from the dictionary. Word and syllable level vector representations were included, according to the method described in [14]. These were learned by taking counts of acoustic events of *f0* and energy stylized by clustered vectors and mean values defined over syllables or

[1] Innoetics: https://www.innoetics.com

[2] IIIT-H: http://speech.iiit.ac.in

words. The training data available for the Challenge was used to learn these matrices. We performed experiments using vectors representations learned over a larger database of a different speaker, but we have observed that results were comparable with speaker-dependent vectors learned on a smaller database.

2.3. Parametric system

As mentioned above, the statistical models used to predict duration and acoustics in our entry are identical to those used for our 2017 entry [2]; details of these models are repeated here for completeness. A conventional two-stage approach was used to predict acoustics: in the first stage, a duration model is used to predict phone durations which are then used to form frame-level linguistic features. In the second stage, an acoustic model is used to generate parameters from those linguistic features.

2.3.1. Feature extraction

Phone sequences were obtained from the text using Festival [15]. Festvox's `ehmm` method [16] was used to modify the phone sequences by the insertion of acoustically-motivated pauses. A state-level forced alignment of these phone sequences with the sentence-segmented audio was then obtained using context-independent HMMs, similar to [17]. Each phone was then characterised by a vector of 481 text-derived binary and numerical features – a subset of the features used as decision-tree clustering questions in the HTS demo [18], adapted for our phoneset.

These questions include linguistic contexts such as quin-phone identity which are added at the phone level, and part-of-speech, positional information relating to syllables, words, phrases, *etc*. All numerical features are given as input (after appropriate normalisation) directly to the network, and not encoded as (for example) 1-of-*k*.

For duration modelling, all these features were used as input and normalised to the range of $[0.01, 0.99]$. The output for training is a five-dimensional vector of durations for every phone, comprising five sub-state durations.

For acoustic modelling, the input uses the same features as duration prediction, to which 9 numerical features were appended. These capture frame position in the HMM state and phoneme, state position in phoneme, and state and phoneme duration, similar to [17].

The speech data was analysed with STRAIGHT [19], and each 5ms frame was represented using 60 mel cepstral coefficients (MCC), measures of aperiodicity in 25 frequency bands (BAP), logarithmic F_0 interpolated through unvoiced regions, and a binary voicing feature. These 87 static features were supplemented with delta and delta-delta features, and for both the duration and acoustic data, a per-component mean and variance normalisation was applied prior to model training, with the transformation reversed as part of synthesis.

2.3.2. Duration model

The duration model trained for our entry to the challenge made use of a simple and straightforward approach with feed-forward neural networks (DNNs) as demonstrated in [20, 21]. The duration model is trained on the aligned data and generates state-level durations given phone-level linguistic features.

The described approach was used only to generated durations, which were then used to form frame-level linguistic features used as input in the generation of acoustic parameters. In contrast to last year's entries, duration predictions are used to compute part of the target cost when performing unit selection (see Section 2.4).

Inputs to the duration model consist of 481 binary and continuously valued features. Its output is a 5-dimensional vector representing state durations in terms of frames. The duration model's architecture consisted of 6 feedforward hidden layers, each with 1024 nodes, using the *tanh* activation function. Mini batch size was set to 64 and learning rate was set to 0.002, being was reduced by 50% with each epoch after the first 10 training epochs.

2.3.3. Acoustic model

The linguistic features extracted from the front-end were converted to numerical vectors using a set of continuous and binary questions [10]. To these, we appended the syllable and word level vector representations based on acoustic counts [14]. The durations generated by the duration model described above were used to propagate all feature to frame-level. These frame-level feature vectors were then used as input to an acoustic model.

A feedforward neural network was trained at the frame-level to map linguistic inputs to vocoder parameters consisting of static and dynamic (delta and delta-delta) features. These acoustic parameters include 60 mel-cepstra coefficients, 25 band aperiodicities, log-f0, and a binary voicing decision. Maximum likelihood parameter generation (MLPG) and postfiltering are then applied to the generated acoustic parameters. In SPSS these parameter trajectories would then be passed through the vocoder to synthesize a speech waveform. Instead, we use them as targets for selecting waveform units (see Section 2.4).

Inputs to the acoustic model consist of the same 481 features used by the duration model. To these, we added syllable and word level vector representations spanning a window of 3 units. To these were appended the nine frame-level features described in [17]. The input vector to the acoustic models consisted of a total of 1900 dimensions. The model consisted of 6 feedforward hidden layers, each with 1024 nodes, using the *tanh* activation function. Mini-batch size was set to 256 and remaining parameters were identical to the duration model.

2.4. Unit selection waveform renderer

For unit selection, we used the halfphone variant of the system described in [4]. The system is summarised here for completeness.

During database preparation, a five-state per phone HMM alignment is used to define halfphone units. This is done by assigning the speech segment corresponding to the first two states as one halfphone (the left halfphone of this phone), and speech corresponding to the last three states as another halfphone (the right). Acoustic features which we call target and join representations are used for computing target and join components of the unit search cost, respectively. The target representation is the concatenation of two streams of acoustic features: a one dimensional value corresponding to the logarithm of fundamental frequency ($\log F_0$) and 60-dimensional vector of STRAIGHT-derived mel cepstral coefficients. For the join representation, we used a 60-dimensional vector of mel-warped log magnitude spectrum extracted pitch-synchronously by the MagPhase vocoder [22], $\log F_0$, and two streams of phase features extracted by MagPhase. We expect that the inclusion of these two phase streams will yield a sequence of speech fragments with fewer phase discontinuities. Standardisation was performed as described in [3, 4]. Previous work [6, §IIB] has show that when

using imperfectly predicted acoustic representations as the basis for a target cost, hybrid systems can compensate for this imperfection if trained with inputs degraded in a consistent way, by regenerating the training data with the model trained on it. Informal experimentation when putting our entry together showed that this helped for the mel cepstral coefficient part of the target representation, but harmed performance when applied to $\log F_0$. We attribute this to the fact that $\log F_0$ is predicted with more variation and less predictability of error. Natural features were therefore used everywhere except MCCs for the target stream, where the regenerated version was used.

As the system operates at the level of the halfphone, the frame-level features described must be mapped to representations at that rate. To obtain halfphone target representations of fixed size we select three frames from the halfphone frame sequence. The first and last are simply the first and last frames in a given halfphone. The middle frame is not necessarily equidistant between those start and end points. Rather, its position is chosen in relation to subphone state boundaries determined during forced alignment, in the expectation that this will provide a more acoustically meaningful point of reference. In practice, we use the last frame of state one as the left halfphone's middle point and the last frame of state four for the right halfphone's midpoint. The halfphone unit's representation is completed by appending a standardised duration of the unit. To obtain the join representation we store frames of join acoustic streams of the start and end of each halfphone. For the unit representation we store references to the start and end samples of the time domain signal. Finally, as well as this numerical data, we store the symbolic phonetic identity of each unit: its quinphone identity and whether it is the left or right halfphone in the phone.

Weighting of the various elements of the target and join representations was performed as described in [3, 4]. Uniform weights were used for the join representations, and weights of 0.1, 0.4 and 0.5 were used for the mel cepstral coefficient, log F0 and duration parts of the target representation, respectively. These weights were determined by limited tuning on some held-out data.

2.5. Speech synthesis

At synthesis time, duration is predicted first, and is used as an input to the acoustic model to predict the speech parameters. Maximum likelihood parameter generation (MLPG) [23] using variances computed from the training data was applied to the output features for synthesis, and spectral enhancement post- filtering was applied to the resulting MCC trajectories. The input to the unit search module consists of phonetic identities, predicted timings and predicted acoustic features which are used to create acoustic 'targets' for unit selection. Concatenation and normalisation of streams is done as in training, using means and standard deviations computed on the training corpus. The halfphones are then resampled in time to a fixed length, consistent with the representations of units in the training database. Viterbi search of the unit database is carried out. We limit the search space by considering a limited number of candidates (50) at each time step. We filtered them according to phonetic type by first taking all units from the database whose quinphone context matches that of the target unit, if any, then do the same for successively more limited contexts: triphone, diphone, and context-independent halfphone, until the desired number of candidates has been selected. In the case of diphone, the direction of context considered depends on whether the target to be matched is the left or right half of a phone. Unit search is treated as a weighted finite-state transducer problem: the target cost is imposed by WFST $\boldsymbol{T}$ and the join cost by $\boldsymbol{J}$. The composition of these produces a WFST whose productions are constrained by both types of cost. The least-penalised path through it is found, corresponding to a sequence of units from the database, whose associated waveform fragments can then be concatenated. Target and join costs are (in effect weighted) Euclidean distances between target and join representations, respectively.

2.6. Concatenation and join smoothing

At unit concatenation time, the time domain signal corresponding to each halfphone is retrieved and analysed on-the-fly with MagPhase [22]. It extracts pitch synchronous speech features in a frame-by-frame basis, describing the complex spectra and F0 contour. The correction/smoothing operations are performed over these features to produce seamless concatenation of units. This process is identical to that used in the 2017 challenge [2], but we repeat a description of it here for completeness.

2.6.1. Concatenation and correction of F0 contours

The $F0$ mid point ($F0_m$) between two consecutive units is given by $F0_m = (F0_p[N_p - 1] + F0_c[0])/2$, where p means preceding unit, c current unit, and N is the unit length in frames.

Then, the slope of the F0 contours of both units are adjusted to reach the $F0_m$ just in the join location. The corrected $F0$ contours are computed by the Equations 1 and 2.

$$F0'_c[n_c] = F0_c[n_c] + (F0_m - F0_c[0]) \cdot \left(\frac{n_c}{1 - N_c} + 1 \right) \quad (1)$$

$$F0'_p[n_p] = F0_p[n_p] + (F0_m - F0_p[N_p - 1]) \cdot \frac{n_p}{N_p - 1} \quad (2)$$

Where $F0'$ is the corrected $F0$, and n is the frame index within each unit. After having all the corrected $F0$ contours for all the units, these are appended building a single $F0$ contour for the whole sentence.

2.6.2. Spectral concatenation and smoothing

Concatenation and smoothing is performed by overlapping and crossfading the complex FFT spectra of two consecutive units. Some extra frames are extracted from the sources, so the units can be overlapped without affecting their expected locations in the synthesised waveform. Three extra frames on each side of the units are extracted from the sources, thus an overlap of seven frames around the joins is produced.

The FFT complex spectrum S is derived from the parameters proposed in [22], M, R, and I, by $S = M \cdot (R + Ij)$. The crossfade is linearly applied to mix the FFT complex spectra of two consecutive units, progressively. It is seven frames length, and in case that a unit is too short, the crossfade is shortened accordingly.

After performing this operation on every join, the FFT complex spectra of all the units are concatenated producing a single complex spectra stream, that describes the whole utterance.

Finally, the signal is synthesised by converting the FFT complex spectra to time domain, and applying Pitch Synchronous Overlap-Add as explained in [22], using the corrected $F0'$ contour.

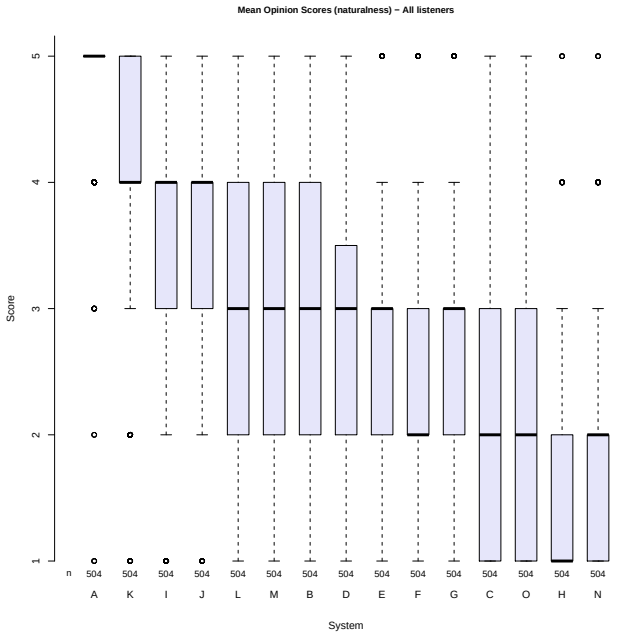

Figure 1: *Our system(L): Mean opinion score for naturalness of the synthesized speech with ratings from all listeners.*

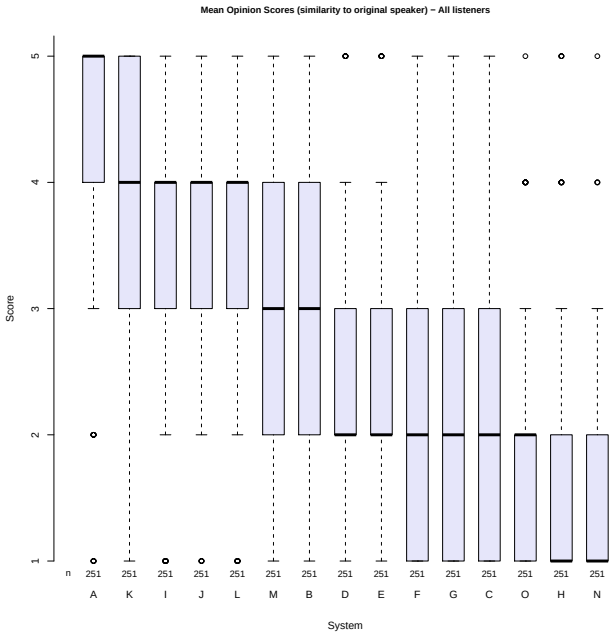

Figure 2: *Our system(L): Mean opinion score for speaker similarity with ratings from all listeners.*

2.7. Paragraph-level synthesis

From the sentences synthesised in this way, files were made containing whole paragraphs, chapters and books as required by the Challenge by simply concatenating the waveforms, exactly as for our entry in the 2016 and 2017 challenges. While proper exploitation of long-distance contexts ought to improve synthesis quality, no contexts outside the current sentence were used for the present submission.

3. Results

The identifier for our system in the published results is L. We base the results discussion presented here on the published statistical analysis of the results that was made at 1% level with Bonferoni corrected alpha [24].

3.1. Naturalness

In Figure 1 we present the mean opinion scores for naturalness from all listeners on book sentences. Our system significantly outperformed two (C and E) out of the four baselines (systems B, C, D and E). Among the 10 other systems participating in the challenge, ours is outperformed only by three (K, I and J). The same trend was observed for results obtained when considering the scores of paid listeners only. For paid participants our system was outperformed only by system K while for volunteers our systems scores were significantly different only from the two best and the two worst systems.

3.2. Speaker similarity

The speaker similarity mean opinion scores from all listeners on book sentences are shown in Figure 2. Considering ratings from all listeners, only system K was significantly better than ours. For the three individual listener groups our system was not outperformed by any other.

3.3. Evaluation of audiobook paragraphs

The results for the evaluation of audiobook paragraphs have been obtained for several factors: stress, intonation, emotion, pleasantness, listening effort, speech pauses and overall impression. We present ratings from all listeners on overall impression in Figure 3. Following the trend observed for the naturalness scores, we note that, in terms of overall impression, our system was significantly outperformed only by systems K, I and J.

3.4. Intelligibility (SUS)

The intelligibility results obtained by our entry were not as positive as other results: only system H was significantly less intelligible than our system. These results show that our system is not as effective on intelligibility as it is on naturalness.

4. Conclusions and future work

For this year's CSTR Blizzard Challenge entry we adopted the same SPSS system used for the hybrid system submitted for last year [2] with a different unit selection component based on the halfphone variant of the system described in [3, 4].

Apart from the intelligibility evaluation, the results obtained by our system are on the whole very positive. There are a few number of potential future improvements which could be made to the hybrid synthesis system described here. These include adopting consistent lexicon-lookup for both the SPSS and unit selection systems, restricting the unit search with a set of higher level acoustic targets and the prediction of phrase breaks.

Reproducibility: We used the Open Source toolkits Merlin[3] for parameter prediction and Snickery[4] for unit-selection.

Acknowledgements: This work was partially supported by: the EPSRC Standard Research Grant EP/P011586/1 *(SCRIPT)*, the EU's H2020 research and innovation programme under the

[3] https://github.com/CSTR-Edinburgh/merlin
[4] https://github.com/CSTR-Edinburgh/snickery

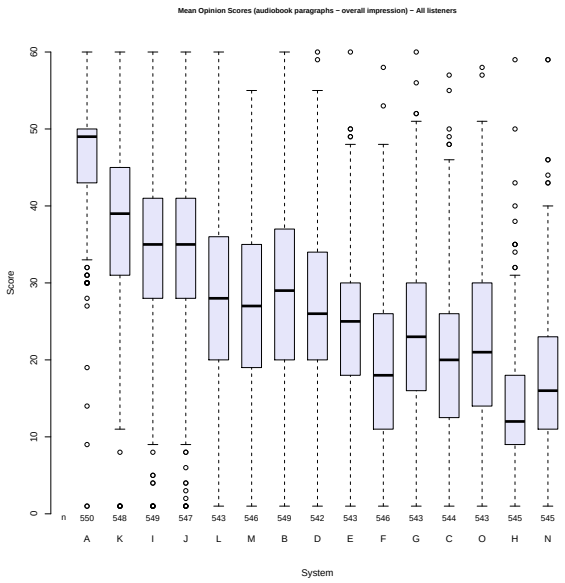

Figure 3: *Our system(L): Mean opinion score for overall impression with ratings from all listeners.*

MSCA GA 67532 *(ENRICH)*, the EPSRC Healthcare Partnerships Programme Grant EP/P02338X/1 *(Ultrax2020)* and by the Chilean National Agency of Technology and Scientific Research (CONICYT) - Becas Chile 72150507.

5. References

[1] T. Merritt, S. Ronanki, Z. Wu, and O. Watts, "The CSTR entry to the Blizzard Challenge 2016," in *Proc. Blizzard Challenge workshop*, 2016.

[2] S. Ronanki, M. S. Ribeiro, F. Espic, and O. Watts, "The CSTR entry to the Blizzard Challenge 2017," in *Proc. Blizzard Challenge workshop*, 2017.

[3] O. Watts, C. Valentini-Botinhao, F. Espic, and S. King, "Exemplar-based speech waveform generation," in *Proc. Interspeech*, September 2018.

[4] C. Valentini-Botinhao, O. Watts, F. Espic, and S. King, "Examplar-based speech waveform generation for text-to-speech," in *IEEE Spoken Language Technology Workshop (submitted)*, 20168.

[5] Z.-J. Yan, Y. Qian, and F. K. Soong, "Rich-context unit selection (rus) approach to high quality tts," in *Proc. ICASSP*, 2010, pp. 4798–4801.

[6] Y. Qian, F. K. Soong, and Z.-J. Yan, "A unified trajectory tiling approach to high quality speech rendering," *IEEE Audio, Speech, Language Process.*, vol. 21, no. 2, pp. 280–290, 2013.

[7] T. Merritt, R. A. J. Clark, Z. Wu, J. Yamagishi, and S. King, "Deep neural network-guided unit selection synthesis," in *Proc. ICASSP*, 2016.

[8] H. Zen, A. Senior, and M. Schuster, "Statistical Parametric Speech Synthesis Using Deep Neural Networks," in *Proc. ICASSP*, 2013, pp. 7962–7966.

[9] G. E. Henter, T. Merritt, M. Shannon, C. Mayo, and S. King, "Measuring the perceptual effects of modelling assumptions in speech synthesis using stimuli constructed from repeated natural speech," in *Proc. Interspeech*, 2014, pp. 1504–1508.

[10] Z. Wu, O. Watts, and S. King, "Merlin: An Open Source Neural Network Speech Synthesis System," in *Proc. SSW*, Sunnyvale, USA, 2016.

[11] O. Watts, A. Stan, R. Clark, Y. Mamiya, M. Giurgiu, J. Yamagishi, and S. King, "Unsupervised and lightly-supervised learning for rapid construction of TTS systems in multiple languages from 'found' data: evaluation and analysis," in *8th ISCA Workshop on Speech Synthesis*, Barcelona, Spain, Aug. 2013, pp. 121–126.

[12] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, 1994, pp. 3–12.

[13] S. Fitt and K. Richmond, "Redundancy and productivity in the speech technology lexicon - can we do better?" in *Proc. Interspeech 2006*, Sep. 2006.

[14] M. S. Ribeiro, O. Watts, and J. Yamagishi, "Learning word vector representations based on acoustic counts," in *Proc. Interspeech*, Stockholm, Sweden, August 2017.

[15] A. Black, P. Taylor, R. Caley, R. Clark, K. Richmond, S. King, V. Strom, and H. Zen, "The festival speech synthesis system, version 1.4. 2," *Unpublished document available via http://www. cstr. ed. ac. uk/projects/festival. html*, 2001.

[16] K. Prahallad, A. W. Black, and R. Mosur, "Sub-phonetic modeling for capturing pronunciation variations for conversational speech synthesis," in *Proc. ICASSP*, 2006, pp. I–853–I–856.

[17] 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*, 2015, pp. 4460–4464.

[18] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. Black, and K. Tokuda, "The HMM-based speech synthesis system (HTS) version 2.0," in *Proc. SSW*, vol. 6, 2007, pp. 294–299.

[19] H. Kawahara, "STRAIGHT, exploitation of the other aspect of VOCODER: Perceptually isomorphic decomposition of speech sounds," *Acoust. Sci. Technol.*, vol. 27, no. 6, pp. 349–353, 2006.

[20] S. Ronanki, Z. Wu, O. Watts, and S. King, "A Demonstration of the Merlin Open Source Neural Network Speech Synthesis System," in *Proc. SSW*, Sunnyvale, USA, 2016.

[21] S. Ronanki, O. Watts, S. King, and G. E. Henter, "Median-based generation of synthetic speech durations using a non-parametric approach," in *IEEE workshop on Spoken Language Technology*, San Diego, California, 2016.

[22] F. Espic, C. Valentini-Botinhao, and S. King, "Direct modelling of magnitude and phase spectra for statistical parametric speech synthesis," in *Proc. Interspeech*, Stochohlm, Sweden, August 2017.

[23] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. ICASSP*, vol. 3, 2000, pp. 1315–1318.

[24] R. A. Clark, M. Podsiadlo, M. Fraser, C. Mayo, and S. King, "Statistical analysis of the Blizzard Challenge 2007 listening test results," *Proc. Blizzard Challenge Workshop*, 2007.