Interspeech 2018
2-6 September 2018, Hyderabad

# Exemplar-based speech waveform generation

*Oliver Watts, Cassia Valentini-Botinhao, Felipe Espic, Simon King*

The Centre for Speech Technology Research, Edinburgh University, UK
owatts@inf.ed.ac.uk, cvbotinh@inf.ed.ac.uk, felipe.espic@ed.ac.uk, Simon.King@ed.ac.uk

## Abstract

This paper presents a simple but effective method for generating speech waveforms by selecting small units of stored speech to match a low-dimensional target representation. The method is designed as a drop-in replacement for the vocoder in a deep neural network-based text-to-speech system. Most previous work on hybrid unit selection waveform generation relies on phonetic annotation for determining unit boundaries, or for specifying target cost, or for candidate preselection. In contrast, our waveform generator requires no phonetic information, annotation, or alignment. Unit boundaries are determined by epochs, and spectral analysis provides representations which are compared directly with target features at runtime. As in unit selection, we minimise a combination of target cost and join cost, but find that greedy left-to-right nearest-neighbour search gives similar results to dynamic programming. The method is fast and can generate the waveform incrementally. We use publicly available data and provide a permissively-licensed open source toolkit for reproducing our results.

**Index Terms**: speech synthesis, vocoder, unit selection

## 1. Introduction

We present a waveform generation module which can be dropped in to a statistical parametric text-to-speech (TTS) synthesis system to turn it into a 'hybrid' synthesiser. By *hybrid*, we mean that waveforms are produced by waveform unit selection and concatenation, but that the selection is guided by the output of a high quality acoustic model. Typically, the acoustic features used to guide selection could themselves be passed through a vocoder to produce a stable, intelligible and reasonably natural-sounding waveform. Until recent developments in the direct time-domain prediction of waveforms, such hybrid systems were the state of the art in natural-sounding speech synthesis, and they are still a dominant form of synthesiser in commercial applications.

See [1, §4] for a review of hybrid approaches where selection is guided by hidden Markov model (HMM)-based synthesis, as well as more recent work where the predictions of neural networks guide selection [2, 3, 4]. In the majority of this work, the speech units selected are relatively large, phonetically determined units, such as diphones and halfphones. The current work aims to use smaller units which can be determined without phonetic annotation. There are several possible benefits to this: it means the unit selection module can be agnostic about the symbolic content of speech to be synthesised in the same way as a vocoder, it opens up the possibility of simply sharing unit databases across dialects and languages, and systems selecting smaller units are conceivably less susceptible to degradation due to inadequate amounts of data and poor annotation. Some work has experimented with small units determined without phonetic alignment – these have always been fixed 5msec frames of speech [5, 6, 7, 8, 9]. However, many of these approaches then use phonetic identity of the segment from which a frame is taken for use in pruning strategies to reduce the computational expense of search. The most similar work to that presented here is [6, 7] where no phonetic annotation is assumed (in the second case due to the need to operate between languages).

Our work is different from most previous work in that we make no reliance on phonetic labels. It differs from all previous work (including [6, 7]) in that no use is made of dynamic programming for unit selection: we find greedy search to be effective. Furthermore, we select units whose temporal bounds are defined by knowledge of speech structure: we select units pitch synchronously rather than using an arbitrary frame size in voiced regions.

It is interesting to consider the system presented here as a generic data-driven waveform reconstruction method. Recent work has shown that statistical models can be trained to predict acceptable sequences of waveform samples either from discrete linguistic features or from intermediate acoustic representations [10, 11, 12]. This latter so-called *neural vocoder* approach – where inputs consist of acoustic features – is similar to exemplar-based waveform generation in that it aims to reconstruct a waveform in a data-driven way given underspecified inputs. Inputs are typically *underspecified* in that phase is missing, and the magnitude spectrum is compressed and simplified to some degree [12, 13]. In these cases, phase and magnitude spectral detail can be restored in a data-driven fashion, as in the case of the exemplar-based method described here. Furthermore, it has been shown [12, §3.3.1] that when working from imperfectly predicted acoustic representations, neural vocoders can compensate for this imperfection if trained with inputs degraded in a consistent way. Similar matched training has been used also in the exemplar-based case [8, §IIB]. As well as benchmarking our system against commonly used vocoders on a simple copy synthesis task, we also explore a variant of such matched training in the current work, by manipulating vocoder parameters to bear some of the characteristics of synthesised speech. This allows us to compare our system's robustness to that of a vocoder synthesise module when processing speech containing TTS-like degradation.

## 2. Proposed system

The task performed by the module at synthesis time is to generate a waveform consistent with a given sequence of feature vectors. As with the inputs to the synthesis module of a standard vocoder, this sequence might be predicted by a statistical model (as part of e.g. a TTS or voice conversion system), or could simply be extracted from natural speech. In all cases, we term this the *target sequence*. For each vector $\boldsymbol{t}'_t$ in the target sequence, an exemplar is chosen by searching a database of units, and finally the exemplars chosen to cover each vector in the tar-

Links to audio samples, code and data for recreating the systems described here can be found at https://github.com/CSTR-Edinburgh/snickery.



10.21437/Interspeech.2018-1857

get sequence are joined to produce a speech waveform. As with other unit selection approaches, the goal of database search is to select a sequence of units to minimise a cost which incorporates two types of constraint. Firstly, each unit should be similar to the acoustics encoded by the target vector (divergence is penalised by the *target* component of the cost); secondly, neighbouring selected units should be acoustically compatible in order to minimise audible artifacts when they are joined (incompatibility is penalised by the *join* component of the search cost). The target and join components can also be thought of as *fidelity* and *fluency* measures: the first scores how faithfully the message encoded by the target sequence is rendered, and the second, how fluently this is done.

### 2.1. Database preparation

The unit database is prepared by acoustically analysing a corpus of (possibly untranscribed) speech. This is done pitch synchronously: analysis starts by placing *pitchmarks* at estimated instants of glottal closure in voiced speech and at 5msec intervals elsewhere. Spectral features characterising the speech around each of these pitchmarks are then obtained, either through pitch-synchronous analysis (as in the experiments reported below, where we use MagPhase [14] to perform analysis pitch synchronously, which in turn relies on REAPER [15] for pitch marking) or by linearly interpolating fixed frame-rate features so that the resulting feature vectors are centred on pitchmarks. We henceforth use the term *frame* to denote pitchmark-centred feature vectors, following [14].

From this analysis, two representations are derived for each pitchmark $i$ for use in search: a target representation $\boldsymbol{t}_i$ and a join representation $\boldsymbol{j}_i$, used for the target and join parts of the search cost, respectively. At time step $t$ when synthesising speech for a novel target sequence, a suitable unit is chosen by considering each unit $i$ in the database and comparing:

1. the target representation $\boldsymbol{t}_i$ of the candidate unit with the target vector $\boldsymbol{t}'_t$ at time $t$ and
2. the join representation $\boldsymbol{j}_{i-1}$ of the unit *preceding* the candidate unit in the database with what can be regarded as the search *history*: the join representation of the unit chosen at synthesis time $t-1$.

Note that the second comparison will be between identical vectors in the case that a unit is considered that is naturally contiguous with the previous unit selected, and so the comparison has the desirable property that the distance between these two vectors will be 0 in such cases.

As these two types of comparison will be made jointly – and units chosen greedily – at each synthesis time step, for convenience a *combined* representation $\boldsymbol{c}_i$ is constructed at database preparation time for each unit $i$ in the database by concatenating the join representation of the preceding unit in the database $\boldsymbol{j}_{i-1}$ with the target representation of unit $\boldsymbol{t}_i$ itself:

$$\boldsymbol{c}_i = [\boldsymbol{j}_{i-1}^\intercal \;\; \boldsymbol{t}_i^\intercal]^\intercal$$

A third type of representation $\boldsymbol{u}_i$ is also stored for unit $i$. This will be concatenated to produce the generated waveform, and might consist of a fragment of time domain signal or some other lossless or high-fidelity representation of that signal. In our experiments, we store and retrieve high-dimensional MagPhase representations of those fragments (in contrast to the low-dimensional MagPhase features used for search), as this allows us to reconstruct the signal at the same time as applying $F_0$ manipulation and spectral smoothing at joins.

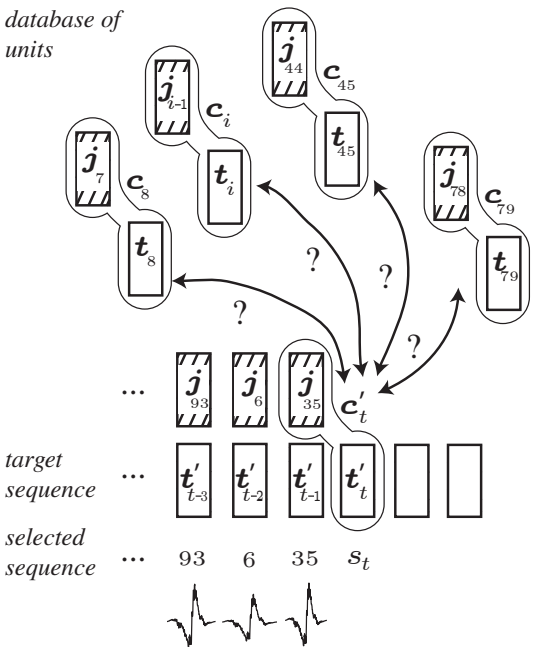


Figure 1: *Exemplar-based synthesis.*

### 2.2. Exemplar-based synthesis

Given a database populated in this way, synthesis proceeds as illustrated in Fig. 1. At synthesis time step $t$, a combined representation $\boldsymbol{c}'_t$ is prepared by concatenating a *history vector* and current target vector $\boldsymbol{t}'_t$. Then the index $s_t$ of the unit selected at time $t$ is determined as:

$$s_t = \arg\min_i \mathcal{D}(\boldsymbol{c}_i, \boldsymbol{c}'_t)$$

where $\mathcal{D}(\cdot,\cdot)$ denotes Euclidean distance. The history vector $\boldsymbol{h}$ is updated to $\boldsymbol{j}_{s_t}$, and search moves to the next timestep $t+1$. On the assumption that any sequence to be synthesised will start with silence, the history vector $\boldsymbol{h}$ is initialised as a join representation of a frame of acoustic silence.

The result of this search is a sequence of indices which allow the retrieval of the portion of acoustics $\boldsymbol{u}_{s_t}$ associated with each selected unit $s_t$ and to generate a waveform. In our implementation this is done after search has finished, but as the search requires no lookahead, in principle the concatenation can be done incrementally as search progresses.

Note that the search is conducted in a greedy fashion: a fixed decision is made at each time step to use the single nearest neighbour in the database. We have experimented also with more conventional dynamic programming solutions, where target cost and join cost are jointly minimised over a lattice constructed using a number of candidate units at each time point. However, informal evaluation suggested that the greedy approach gives comparable results, and as it also has the benefits of simplicity and lack of look-ahead, we have decided to focus exclusively on this approach in the present work. We suppose that greedy search is sufficient in our case as the fragments selected by our system are not long enough that feature trajectories can stray far from the target sequence in the course of a single unit.

### 2.3. Generalisation to multiple epochs at each timestep

So far we have considered the case where a single epoch is selected at each synthesis timestep, i.e. the waveform is generated

by concatenating 1-epoch fragments. While this results in intelligible speech, we have obtained better results by selecting an $m$-epoch chunk of speech at each timestep, where $m > 1$. The following changes are made to database preparation and synthesis routine to achieve this. The combined representations $\boldsymbol{c}_i$ of database unit $i$ is obtained as follows:

$$\boldsymbol{c}_i = [\boldsymbol{j}_{i-m}^\intercal \;\; \boldsymbol{t}_{i-m+1}^\intercal \;\; \ldots \;\; \boldsymbol{t}_i^\intercal]^\intercal$$

Note that most epochs therefore appear in $m$ combined representations (i.e. the longer units defined in this way overlap temporally in the training database).

At synthesis time, the combined representation $\boldsymbol{c}'_t$ at time $t$ when simultaneously selecting $m$ epochs is prepared by concatenating the history vector $\boldsymbol{h}$ and target vectors $\boldsymbol{t}'_{t-m} \ldots \boldsymbol{t}'_t$. At each iteration, the history $\boldsymbol{h}$ is then updated to $\boldsymbol{j}_{t_s}$ and the time index is then incremented from $t$ to $t + m$. That is, contiguous sections of speech consisting of $m$ epochs are concatenated in a non-overlapping fashion, and the join comparison is made between single frames of join representation features at unit boundaries, as illustrated in Fig. 2 for $m$=3.

Typical settings of $m$ we have used are between 2 and 8. In effect, this corresponds to selecting longer units from the database, and there will be at most a single concatenation every $m$ synthesised pitchmarks. While both the greedy and dynamic programming methods with which we have experimented in principle allow such sequences of units to be selected, in practice we have found that explicitly adding the constraint that each non-overlapping subsequence of $m$ units should be selected contiguously results in the discovery of perceptually better unit sequences.

### 2.4. Target and join representations

So far, nothing specific has been said about the representations used for target and join components of the cost. Typically, several 'streams' of acoustic features will be used to build both the target and join representations for a unit. The target representation must contain enough detail to guide the selection of appropriate fragments of speech, and the join representation must contain enough information to determine whether a given pair of units can be combined smoothly. While the two representations could in principle be identical, an important distinction is that the join representations of natural units in the database will only ever be compared with other natural units, whereas the target representations stored in the database can be compared with synthetic target sequences. This suggests that more compact representations, and ones which average more sensibly, might be preferred as target representation. It also suggests that performance might benefit from corrupting the database target representations in a way that is consistent with the kind of corruption envisaged at synthesis time (see Section 3).

In the experiments presented here, we use logarithm of fundamental frequency ($\log F_0$) and 60-dimensional mel-warped log magnitude spectrum extracted pitch-synchronously by the MagPhase vocoder [14] as streams in our target representation, but many other choices are possible. For the join representation, we supplement those streams with the two streams of phase features extracted by MagPhase. Note that while our target representation contains no phase information, the inclusion of phase in the join representation means that we expect unit selection to yield a sequence of speech fragments which are compatible in terms of phase. In turn this means that any TTS system whose predictions are used to guide unit selection need not output phase information (cf. WORLD [16]).

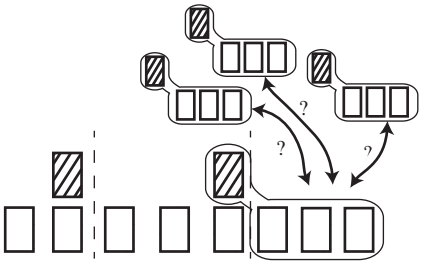


Figure 2: *Generalisation to multiple epochs.*

#### *2.4.1. Standardisation*

The separate streams' features are standardised, weighted and concatenated to yield either target or join representations. When a stream of features is standardised, means over the whole database are computed per coefficient so that the standardised values will all have zero mean, but a single standard deviation value is used to scale all coefficients in each stream. This is motivated by the assumption that each of the streams we have chosen to use has been designed in such a way that the relative dynamic range of coefficients in a stream is proportional to their relative perceptual importance, and we wish to preserve these difference of range in the standardised values.

Unvoiced frames of $F_0$ and $\log F_0$ are ignored when computing means and standard deviations. Unvoiced values are also treated specially when streams are standardised – they are assigned a negative value whose magnitude is given by multiplying the feature's standard deviation by a constant factor (set to 20). The motivation here, following [17] is that we wish to: penalise differences between voiced values as normal but also place no $F_0$ penalty on comparisons between unvoiced features and place a large penalty on comparisons between voiced and unvoiced features.

#### *2.4.2. Weighting*

Features' contributions to selection costs can be modified by weighting. Weights are applied stream-by-stream rather than coefficient-by-coefficient. This is both to reduce the number of parameters which must be manually adjusted, and also follows the logic outlined in Section 2.4.1: we expect streams to have been engineered in such a way that the component coefficients' relative dynamic ranges (and therefore their contribution to a Euclidean distance) reflect relative perceptual importance.

We configure the stream weights as follows. Firstly, for both target and join costs, we set stream weights so that they sum to 1. The join representation is then scaled globally by a factor $\alpha$, where $0 < \alpha < 1$, and the target representation is scaled by $1 - \alpha$. As in other approaches to unit selection, this allows us to strike the right balance between fidelity and fluency.

## 3. Experiments

We present the results of an experiment where the conditions summarised in Table 1 were compared side by side.

The experiment has two goals; firstly, we wish to benchmark the quality of speech synthesised by the proposed system against the speech reconstructed by two freely-available vocoders in a simple copy synthesis task (i.e. where vocoder features and unit selection target features are extracted from natural speech and used directly). This can be done by comparing conditions W0, M0 and S0.

Secondly, although we leave the integration of our waveform generator into a full TTS system for future work, we wish

Table 1: *Experimental conditions.*

| System | Description | Smoothing |
|---|---|---|
| N | Natural speech | None |
| W0 | Vocoded speech WORLD [16] | None |
| M0 | Vocoded speech MagPhase [14] | None |
| S0 | Proposed system | None |
| M1 | Vocoded speech MagPhase [14] | slight |
| S1 | Proposed system | slight |
| M2 | Vocoded speech MagPhase [14] | extreme |
| S2 | Proposed system | extreme |

to ascertain how robust the proposed method is to degradations of the input data of the sort which would be expected in a TTS system. For this purpose, target features were degraded in a way consistent with the effects of prediction from text; following [18], degraded features were created by smoothing the original target features with a 5-frame Hanning window followed by scaling their standard deviations to either 80 % (slight smoothing) or 60 % (extreme smoothing) or the original. To limit the number of conditions to be compared to manageable numbers, this processing was applied to features from only one of the two vocoders used (MagPhase).

### 3.1. Database, proposed system and baselines

We used a dataset of recordings of the speech of a male native English speaker sampled at 48 kHz. To construct the proposed system we used 2004 sentences (containing 83 minutes of audio) for creating the unit database and 19 other sentences for some limited system tuning. This results in a database with approximately 750,000 units. A single configuration of the system was used for all of conditions S0, S1 and S2 as follows. Uniform stream weights were used for the join and target representations, but the influence of the join component of the cost was reduced by setting $\alpha$ to 0.2. The number of frames selected simultaneously $m$ was set to 6, and selected units were extended by 1 frame on either side to allow 2 frames of cross-fade to be applied to the spectral representations from which speech is resynthesised. The target $F_0$ was imposed on selected units when speech was resynthesised with MagPhase. The combined representations of our database were indexed with a $k$-d tree for efficient approximate search. Although our implementation is in many ways naive and we leave efficiency improvements for future work, we are able to synthesise speech faster than real time on a large memory server (excluding the overhead of loading the model at the start of a synthesis session).

We created the WORLD vocoder [16] baseline by using the version of the vocoder distributed with the Merlin toolkit [19]. With it we extracted 60 Mel cepstral coefficients and 5 band aperiodicities; $F_0$ was extracted using REAPER [15]. WORLD's synthesis module was then used to reconstruct speech from these three acoustic streams.

The MagPhase baseline was created using the MagPhase vocoder implementation released in [14]. Using this we extracted 60 magnitude, 45 imaginary and 45 real features. $F_0$ was extracted using REAPER [15]. Speech was reconstructed from these four acoustic features.

### 3.2. Listening experiment design

We conducted a MUSHRA-style test [20] with 21 screens. On each screen participants could play the audio produced by different systems for the same sentence. Listeners were asked to rate the quality of the samples on a scale from 0 (bad) to 100 (excellent). The first screen was used only for training participants. A different sentence was used for each screen. Across every five participants, 100 different sentences were used. Natural speech was included on each screen so that participants would have a quality reference and to check if participants paid sufficient attention to score it as 100 as instructed. We recruited 20 native English speakers. We excluded 13% of screens where listeners did not give natural speech (the reference) the highest score (as instructed). Three participants were excluded who rated N less than 100% in more than 25% of screens.

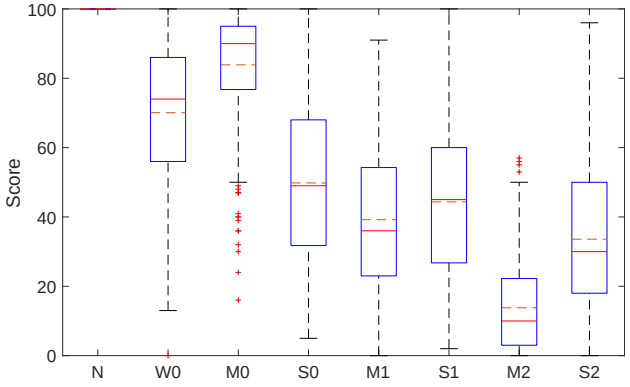


Figure 3: *Boxplot of listening test scores.*

### 3.3. Results

A boxplot of the results is presented in Fig. 3. Solid and dashed lines indicate the median and mean values of each distribution. To test if differences were significant, we used a Mann-Whitney U test, at a p-value of 0.05, and with a Holm Bonferroni correction due to the large number of pairs to compare. All systems were perceived to be significantly different from each other.

As illustrated in Fig. 3, MagPhase (M0) obtained the highest mean scores, followed by World (W0) and the proposed system (S0). This result is unsurprising as WORLD – in contrast to MagPhase – relies on the minimum phase assumption. Results of the smoothed conditions (1,2) show a different trend. When features are smoothed, MagPhase scores suffer considerably more than our system's, to the extent that our system outperforms the vocoder in the degraded conditions.

## 4. Conclusions

We have proposed a new method for generating speech waveforms from low-dimensional target acoustic representations. To generate speech we select small units from a database in a greedy fashion which enables incremental generation. Unlike a vocoder, our method has the potential to compensate for the kind of degradation observed in acoustic representations predicted by statistical models. In contrast to most exemplar-based systems, our method does not require data with phonetic level annotation as units are not defined by phone boundaries, but by pitchmarks automatically extracted from speech. We observe that when the low-dimensional representation is derived from natural speech directly, our method is outperformed by two state-of-the-art vocoders; however, the performance of our method degrades more gracefully as the amount of imperfection imposed on the inputs features is increased. We believe results can be improved by using representations that average more sensibly, and by much more careful condition-dependent tuning of our system.

**Acknowledgements:** This research was supported by EPSRC Standard Research Grant EP/P011586/1. Thanks to three anonymous Interspeech reviewers and to Dr. Gustav Eje Henter for helpful comments on an earlier version of this paper.

# 5. References


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

[2] T. Merritt, R. A. Clark, Z. Wu, J. Yamagishi, and S. King, "Deep neural network-guided unit selection synthesis," in *Proc. ICASSP*, 2016, pp. 5145–5149.

[3] T. Capes, P. Coles, A. Conkie, L. Golipour, A. Hadjitarkhani, Q. Hu, N. Huddleston, M. Hunt, J. Li, M. Neeracher, K. Prahallad, T. Raitio, R. Rasipuram, G. Townsend, B. Williamson, D. Winarsky, Z. Wu, and H. Zhang, "Siri on-device deep learning-guided unit selection text-to-speech system," in *Proc. Interspeech*, August 2017.

[4] V. Wan, Y. Agiomyrgiannakis, H. Silen, and J. Vit, "Google's next-generation real-time unit-selection synthesizer using sequence-to-sequence LSTM-based autoencoders," in *Proc. Interspeech*, August 2017.

[5] Z. Ling and R. Wang, "HMM-based unit selection using frame sized speech segments," in *Proc. Interspeech*, 2006.

[6] T. Hirai, J. Yamagishi, and S. Tenpaku, "Utilization of an HMM-based feature generation module in 5 ms segment concatenative speech synthesis," in *Proc. SSW*, August 2007.

[7] Y. Qian, J. Xu, and F. K. Soong, "A frame mapping based HMM approach to cross-lingual voice transformation," in *Proc. ICASSP*, May 2011, pp. 5120–5123.

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

[9] Z. P. Zhou and Z. H. Ling, "DNN-based unit selection using frame-sized speech segments," in *Proc. Int. Symp. on Chinese Spoken Lang. Proc.*, Oct 2016, pp. 1–5.

[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." [Online]. Available: http://arxiv.org/abs/1609.03499v2

[11] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, and Y. Bengio, "SampleRNN: An Unconditional End-to-End Neural Audio Generation Model." [Online]. Available: http://arxiv.org/abs/1612.07837v2

[12] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions." [Online]. Available: http://arxiv.org/abs/1712.05884v2

[13] A. Tamamori, T. Hayashi, K. Kobayashi, K. Takeda, and T. Toda, "Speaker-dependent WaveNet vocoder," in *Proc. Interspeech*, August 2017, pp. 1118–1122.

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

[15] "REAPER: Robust Epoch And Pitch EstimatoR," https://github.com/google/REAPER, 2017.

[16] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: A Vocoder-Based High-Quality Speech Synthesis System for Real-Time Applications," *IEICE Trans. Inf. Syst.*, vol. E99.D, no. 7, pp. 1877–1884, 2016.

[17] R. A. Clark, K. Richmond, and S. King, "Multisyn: Open-domain unit selection for the festival speech synthesis system," *Speech Communication*, vol. 49, no. 4, pp. 317 – 330, 2007.

[18] T. Merritt, J. Latorre, and S. King, "Attributing modelling errors in HMM synthesis by stepping gradually from natural to modelled speech," in *Proc. ICASSP*, April 2015, pp. 4220–4224.

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

[20] *Method for the subjective assessment of intermediate quality level of coding systems*, ITU Recommendation ITU-R BS.1534-1, International Telecommunication Union Radiocommunication Assembly, Geneva, Switzerland, March 2003.