HAL
open science

Hider-Finder-Combiner: An Adversarial Architecture for General Speech Signal Modification

Jacob J Webber, Olivier Perrotin, Simon King

To cite this version:

Jacob J Webber, Olivier Perrotin, Simon King. Hider-Finder-Combiner: An Adversarial Architecture for General Speech Signal Modification. Interspeech 2020 - 21st Annual Conference of the International Speech Communication Association, Oct 2020, Shanghai (Virtual Conf), China. 10.21437/interspeech.2020-2558 . hal-02987956

HAL Id: hal-02987956

https://hal.science/hal-02987956v1

Submitted on 4 Nov 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Hider-Finder-Combiner: An Adversarial Architecture For General Speech Signal Modification

Jacob J Webber[1], *Olivier Perrotin*[2] *and Simon King*[1]

[1]The Centre for Speech Technology Research, University of Edinburgh, United Kingdom
[2]Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, France
j.j.webber@ed.ac.uk

Abstract

We introduce a prototype system for modifying an arbitrary parameter of a speech signal. Unlike signal processing approaches that require dedicated methods for different parameters, our system can – in principle – modify any control parameter that the signal can be annotated with. Our system comprises three neural networks. The 'hider' removes all information related to the control parameter, outputting a hidden embedding. The 'finder' is an adversary used to train the 'hider', attempting to detect the value of the control parameter from the hidden embedding. The 'combiner' network recombines the hidden embedding with a desired new value of the control parameter. The input and output to the system are mel-spectrograms and we employ a neural vocoder to generate the output speech waveform. As a proof of concept, we use F_0 as the control parameter. The system was evaluated in terms of control parameter accuracy and naturalness against a high quality signal processing method of F_0 modification that also works in the spectrogram domain. We also show that, with modifications only to training data, the system is capable of modifying the 1st and 2nd vocal tract formants, showing progress towards universal signal modification.
Index Terms: speech synthesis, adversarial networks, speech modification

1. Introduction

Current speech synthesisers are typically [1] separated into two main components: the sequence-to-sequence model converts sequences of graphemes or phonemes into sequences of frames in the frequency domain, then a neural vocoder generates a waveform. These systems yield high quality results, but are limited to synthesis of voices for which audio exists. This is true even when using *adaptation*, by injecting high-level information (speaker identity [2], speaking style [3], expressivity, etc.) into the model. Whilst current systems have impressive *adaptation* ability [4], they are driven by data and generally offer no explicit *control* of speech parameters such as pitch or timbre. Some recent neural vocoders [5, 6] accept an explicit pitch parameter as input but at the expense of speech quality [7].

Using signal processing, manipulation of arbitrary speech parameters is challenging, not least because speech parameters co-vary. However, using signal processing only to extract parameters – to *annotate* waveforms – is far less challenging.

The limitations of current neural approaches, and the relative ease of annotating speech compared to signal processing manipulation, together motivate the approach presented here. We aim for true *controllability* of speech. This is performed in the mel-spectrogram domain because of its ability to represent most aspects of speech and its widespread use as the interface between sequence-to-sequence models for speech synthesis neural vocoders [8, 9]. We present a proof of concept for a machine learning-based approach to modification of speech signals for any arbitrary *control parameter*, which requires only a training dataset *annotated* with this control parameter. The method aims to modify precisely *one control parameter* whilst leaving all other aspects of the speech signal unchanged.

The first author is funded by the Engineering and Physical Sciences Research Council (grant EP/L01503X/1), EPSRC Centre for Doctoral Training in Pervasive Parallelism at the University of Edinburgh, School of Informatics.

2. Controllability

2.1. What is Ideal Control?

We can annotate many parameters on a speech signal; e.g., F_0 (perceived as pitch), spectral tilt (voice quality), or formant positions (articulation) can be estimated automatically from waveforms. We could imagine annotation of other parameters using external information (e.g., physical articulator positions) or human perception (manual labelling). Ideal controllability implies being able to modify one parameter while changing as few other parameters as possible. However, many parameters are deeply intertwined with others. For example, mel-spectrograms depend on (at least) F_0, speaker characteristics and linguistic content. Ideal control of F_0 means changing only that part of the mel-spectrogram which is dependent on F_0.

2.2. Signal Processing for Speech Modification

Vocoders achieve general-purpose speech parameter manipulation by decomposition of the signal into source and spectral-envelope features. For example, STRAIGHT [10] or WORLD [11] use smooth spectral envelope features [12] which allow spectral modification independently from F_0 and duration. Nevertheless, they are limited to controlling parameters which can be mapped to/from those speech features. GFM-IAIF [13] represents the spectral envelope with vocal tract and glottis-related parameters, for greater controllability [14]. Non-parametric methods avoid decomposition of the signal but are generally restricted to modifying one parameter. For instance, TD-PSOLA [15] and phase vocoders [16] are capable of high-quality F_0 and duration modification, but do not generalise to other control parameters. Although such methods are effective, modifying one parameter often fails to preserve its natural co-variation with other parameters, resulting in artefacts.

2.3. Controllability and Machine Learning

Our proposed system uses machine learning techniques to modify a speech signal by an arbitrary *control parameter*. Conventional neural networks are learned *universal function ap-*

Copyright © 2020 ISCA

http://dx.doi.org/10.21437/Interspeech.2020-2558

proximators. Consider the function f that maps input mel-spectrogram x to output mel-spectrogram z and approximate it with a network F trained to minimise error $\epsilon = |f(x) - F(x)|$ averaged over all (x, z) in some training set. If we add another control parameter y as input to the network, so that $z \approx F(x, y)$, F will simply learn to ignore y if x contains sufficient information to predict z. Even if x contains only partial information, when we attempt to control the speech z by varying control parameter y we risk contradicting information contained in x. To solve these related problems, we employ a hidden representation h which contains as little information as possible about y, but all *other* information from x that, when combined with y, will predict z. We use a *hider* (H) and *combiner* (C) network for this, with a *finder* (D) adversary.

$$h = H(x) \quad (1)$$
$$z \approx C(h, y) \quad (2)$$
$$y \approx D(h) \quad (3)$$

This is similar to a *Generative Adversarial Network* (GAN), in which a generator network aims to fool a discriminator (adversary) network into misclassifying its output, e.g., as being genuine [17]. Generator and discriminator are trained in turn. In our system, the *finder* is the discriminator.

3. The Proposed Architecture

Fig. 1 shows the architecture, and [18] describes how this arose out of our earlier experiments:

1. The *hider* takes as input a mel-spectrogram and produces as output a 2-dimensional hidden embedding (one axis is time, as in the spectrogram) which preserves sufficient information for reconstructing the spectrogram but as little information about the control parameter as possible.
2. The *finder* network is an adversary that attempts to detect the value of the control parameter from the output of the *hider*. It is only used to train the system and is not required during generation.
3. The *combiner* network takes as input the hidden embedding and a new value of the control parameter and produces as output a mel-spectrogram.

In the following, we use F_0 as an example control parameter. The architecture is trained from scratch without any supervision of the individual networks, using only ground-truth mel-spectrograms paired with corresponding control parameter values; input and output are identical in training. The goals of training are (1) when given the original value of the control parameter, the combiner output is the same as the hider input, and (2) the *finder* cannot detect the value of the control parameter from the *hider* output.

Hider: Input ground-truth mel-spectrograms are passed through a fully-connected layer, into a convolutional layer of kernel size 10, then a 3-layer 800-node GRU with tanh activation. The features are resized to the correct dimensionality using a fully-connected layer.

Finder: Hidden embeddings are passed directly into a 2-layer, 300-node stacked GRU with tanh activation, then one linear layer, and finally a softmax layer which produces a probability distribution across the quantised F_0 values.

Combiner: The architecture of the combiner is shown in Fig. 2. Inputs are a voicing flag, the control parameter, and the output of the hider network. The combiner first passes a

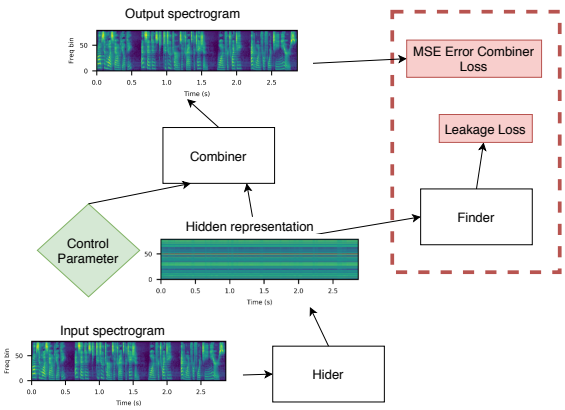

Figure 1: *The complete architecture. Components in the dashed box are used only during training.*

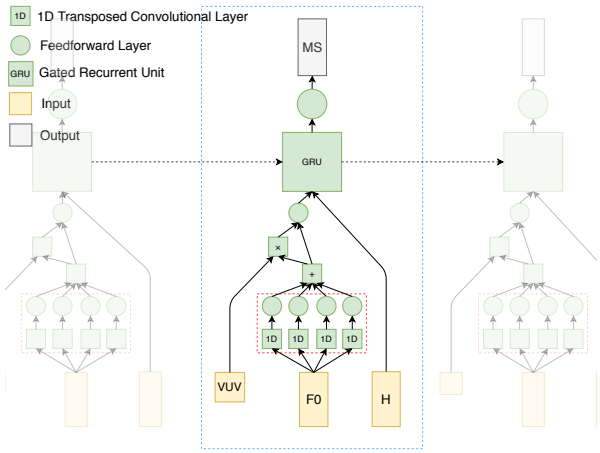

Figure 2: *The combiner network architecture across three time frames. The blue dotted box shows one frame. The red dotted box shows a parallel bank of transposed convolutions (PBTC). H is the output of the Hider network, VUV is a voiced/unvoiced flag, F_0 is the control parameter.*

1-hot embedding of the control parameter through a parallel bank of transposed convolutions (PBTC). A PBTC consists of an array of 1D transposed convolutional layers, with each of these convolutions using a different dilation and the results being summed together. The aim of this PBTC is to generate the harmonic structure in the spectrogram characteristic of F_0. Because convolutional networks are powerful, this approach generalises to other features where a single control parameter results in structured spectral energy. The output of the PBTC is duplicated, with one of the two copies being masked by voicing (zeroed out in unvoiced regions). This duplication allows the system to learn whether to take into account voicing information, and was found to reduce artefacts in unvoiced sections of speech. For the PBTC, 10 dilated transposed convolutional layers were used with kernel size 50 and dilations in the range 2-20. The result of the PBTC is concatenated with the hidden embeddings coming from the hider, and passed to a 3-layer 1200-node stacked GRU with tanh activation. Finally, a fully-connected layer outputs a mel-spectrogram.

3.1. End-to-end Adversarial Training

The control parameter in initial experiments was F_0. Training data was automatically annotated for F_0 with WORLD's pitch

tracker [11] and interpolated through unvoiced sections, then quantised into 80 linear-scale bins between 60 Hz and 500 Hz. We refer to this vector as 'ground truth F_0'. Similar to GANs, a two stage process is used to update models at each training step. First, the *finder* (adversary) is trained. For each utterance, the input mel-spectrogram is passed through the *hider*. It outputs a hidden embedding that is used to train the *finder*, with ground truth F_0 as target, to minimise cross-entropy loss. Second, the hider and combiner networks are jointly trained by backpropagating through both networks to minimise an adversarial loss defined as the weighted sum of the *combiner loss*, $\mathcal{L}_{\text{combiner}}$ and the *leakage loss*, $\mathcal{L}_{\text{leakage}}$.

$$\mathcal{L}_{\text{adversarial}} = \mathcal{L}_{\text{combiner}} + \beta \cdot \mathcal{L}_{\text{leakage}} \tag{4}$$

The MSE *combiner loss* is measured at the output of the combiner, whose target is the same as the input to the hider: a ground-truth mel-spectrogram. The combiner also takes ground-truth F_0 as an input.

The leakage loss measures how much information related to the control parameter is 'leaked' by the hider into the hidden embedding. Defining a representation that contains *no* information pertaining to a specific variable is non-trivial. *Mutual information* quantifies the amount of information that one random variable contains about another and this would be applicable to neural networks, which map from one random variable to another. However, calculating mutual information is difficult and can require a large number of training samples [19].

We used a simpler solution: the hider network should produce a hidden embedding from which the finder outputs a uniform probability distribution over the quantised control parameter (CP): it should be *maximally uncertain* about the CP. The MSE error between finder output and a uniform distribution is equivalent to the variance of the distribution and thus the leakage loss was defined. A high value implies the finder network has certainty about the CP. Zero loss implies that all CP values are equally probable, because the hider has completely removed all information about the CP from the hidden embedding. Maximal uncertainty at finder output is preferable to a confident-but-incorrect finder. In the latter case, the hider could learn to fool the finder network with a solution that is easier to learn than actually hiding the CP – perhaps, where F_0 is the CP, by doubling F_0 via only removing odd harmonics – but that the combiner could use instead of its control parameter input value.

3.2. Experimental Setup

The LJSpeech corpus [20] was used for training, comprising recordings of 50 chapters of non-fiction books by a female US English speaker. The last chapter was held out for the evaluation. Batch size was 1 utterance with 12693 training steps per epoch. Parameters were updated using the Adam optimizer [21] after every training step. A validation set of 128 randomly-selected utterances was used for early stopping and all systems reported below were trained for 6 epochs. For the input and output, as in [8], a frame size of 50 ms and a frame shift of 12.5 ms are used to extract an 80-band mel-spectrogram with LibROSA [22]. A feature width of 80 was also selected for the hidden embedding. The audio sampling rate was 22.05 kHz.

The value of β in eqn. 4 determines how much the leakage of control parameter value into the hidden embedding contributes to training. At $\beta = 0$ there is no adversarial loss, the hidden embedding will contain control parameter information, and the combiner is free to ignore its input control parameter. Small non-zero values of β lead to partially-modified speech, as

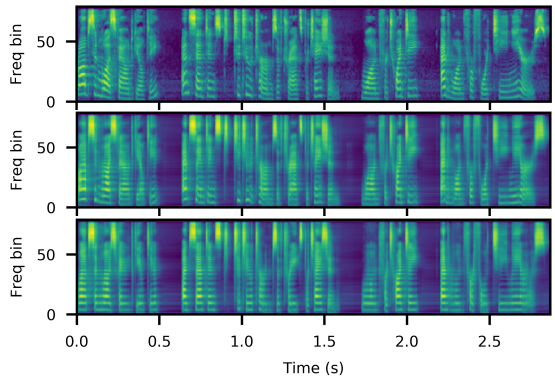

Figure 3: *Example of changing F_0 to a constant value of* 214 Hz. *Top: ground truth mel-spectrogram. Middle: output with $\beta = 200$. Bottom: output with $\beta = 800$.*

shown in Fig. 3. Increasing β brings the output speech closer to the desired control parameter value, (bottom of Fig. 3) but excessively large values of β cause output quality to degrade because too little importance is given to the combiner MSE loss. Overall, the MSE combiner loss is positively correlated with β and the leakage loss is negatively correlated. Pilot experiments [18] showed that $\beta = 560$ is a satisfactory trade-off between output quality and control accuracy, and this value is used in the following evaluation.

4. Evaluation

The system was evaluated with F_0 as the control parameter (CP). We selected 30 utterances from the test chapter, and defined four control types: the original F_0 trajectory (copy); scaled F_0 by -50% to +50% of its original value in steps of 10; entirely new, but plausible, F_0 contours generated through human performance ('drawn') as in [23], with average value set to match the original F_0. As one baseline, we used a digital signal processing (DSP) approach that also operates on mel-spectrograms. All audio was generated from the baseline and the proposed Hider-Finder-Combiner approach (HFC) mel-spectrograms by an open-source implementation of a state-of-the-art neural vocoder [9, 24]. As a second baseline, we used WORLD, with speech waveforms as input and output. We provide an objective evaluation of how closely the output of our system follows the specified control parameter value, and a subjective evaluation of the audio quality.

4.1. The Baseline DSP System

To disentangle the harmonics of F_0 and vocal tract (VT) resonances (formants), in order to allow F_0 modification without changing formants, source-filter separation is performed with GFM-IAIF [13] on mel-spectrograms. This decomposes the spectral envelope into two sets of linear predictive (LP) coefficients [25], modeling the glottis and the VT respectively. F_0 scaling is applied to the excitation signal obtained by inverse filtering the speech frame with the glottis and VT filters. This modification follows the principle of the phase vocoder [16], modified slightly to use mel-spectrograms: since phase reconstruction is done by the neural vocoder, the DSP system only performs frequency-stretching or compressing on the amplitude spectrum by a rescaling factor. Also, each mel-spectrogram frame is interpolated to a linear frequency scale before F_0 modification, then mel-scaled again afterwards. Finally, the speech

	WORLD	DSP	HFC
F_0 copy	0.16	0.16	0.16
F_0 scale	0.18	0.20	0.23
Drawn F_0	0.15	0.18	0.14

	GFM-Voc	DSP	HFC
F_1	0.33	0.26	0.36
F_2	0.32	0.34	0.41

Table 1: *RMSE (in octave) of* $\log_2(F_0)$, $\log_2(F_1)$ *and* $\log_2(F_2)$ *between control and synthesis for the models in consideration.*

frame is reconstructed by filtering the F_0-scaled source signal with both glottis filter and VT filters before going back to the mel-frequency scale to feed the neural vocoder.

4.2. Objective Evaluation

To assess the fidelity of the output to the control parameter, we extracted F_0 from the synthesised speech waveforms with WORLD and measured the root mean square error (RMSE) with their respective F_0 control values, on a logarithmic scale. Table 1 (top) shows the median values of the $\log_2(F_0)$ RMSE distributions for each system and F_0 modification type. A Kruskal-Wallis rank-sum test using a χ^2 distribution shows that the RMSE differences between models are not significant for copy (χ^2 = 1.2, $p = 0.55$), are significant for scale (χ^2 = 49.3, $p < 1e^{-10}$) and marginally significant for drawn F_0 (χ^2 = 7.2, $p < 0.03$). A post-hoc Dunn test for pairwise comparison between methods assesses that all measures are different for scaled and drawn F_0 ($p < 0.01$), except between HFC and WORLD for drawn F_0 ($p = 0.5$). The proposed system performs as well as the others for copying the F_0 trajectory. For F_0 scaling, there is significant but small difference, with an RMSE of 0.02 octave higher for HFC than the DSP method. For drawn F_0, HFC performs significantly better than DSP, and as well as WORLD. From these results, it appears that HFC is better for arbitrary modification of F_0 (drawn F_0) than the DSP method. The superiority of the DSP method for the narrow case of F_0 *scaling* is not surprising, given the simplicity of stretching mel-spectrograms of the glottal source signal.

4.3. Subjective Evaluation

To assess naturalness, a comparative mean opinion score (CMOS) evaluation was performed. Listeners were asked to compare utterances in pairs on a five point scale, corresponding to whether they thought either option sounded much more natural, slightly more natural, or whether both options sounded similarly natural. Each pair comprised the same utterance generated with DSP and HFC. From our speech material, we selected synthesis with original F_0 (copy); scalings of $\pm 20\%$ from the original F_0, and the drawn F_0, leading to a total of 120 pairs that were randomly split into two blocks. 50 native English speaking US citizens were recruited and paid using the Prolific Academic platform. 25 listeners judged block 1, and 25 others judged block 2. An equal number of pairs was presented in each order (DSP-HFC, HFC-DSP) and the ordering of pairs was randomised per listener.

Fig. 4 shows the CMOS for the three different conditions. A Kruskal-Wallis rank-sum test followed by a Dunn test for pairwise comparison shows that the *drawn* distribution is significantly different from the two others ($p < 1e^{-15}$), and that the *copy* and *scale* distributions are marginally significantly different ($p < 0.03$). This evaluation shows that HFC is as good as DSP for *copy* and *scale*, and better for the generation of new F_0

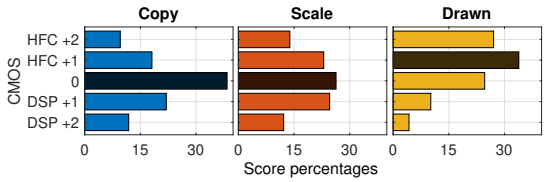

Figure 4: *CMOS between DSP and HFC for different* F_0 *modifications. The dark bars indicate the median score for each condition.*

trajectories. These results are consistent with the objective measures, and confirm that HFC is better for the most practically-useful case of arbitrary modification of F_0.

5. Towards a Universal Signal Modifier

Our approach is intended to generalise to the modification for any control parameter which can be annotated on speech waveforms. We now demonstrate this generality by modifying the first or second formant (F_1/F_2), which we annotated on training data using the GFM-IAIF VT filter [13]. F_1 and F_2 are significantly more challenging to manipulate than F_0. As with F_0, the control parameter was quantised into 80 linearly spaced bins: F_1 between 0.2 and 1 kHz, F_2 between 0.5 and 3 kHz.

We synthesised the 50 sentences of median length from the test chapter in which we scaled either F_1 or F_2 by between -40% and $+40\%$ of their original value in steps of 20. Both HFC and DSP methods were compared against GFM-Voc [14], which extracts the VT filter with GFM-IAIF, changes the formant position through pole modification, then filters the unchanged source signal to reconstruct the speech waveform. This is the same process that is applied in DSP, except that in the former, the waveform is obtained by inverse Fourier transform of the spectrogram, using the original phase, while in the latter the mel-spectrogram is fed to the neural vocoder. Since GFM-IAIF is used for both data annotation and parameter modification in DSP, we used Praat [26] for independent extraction of F_1 and F_2 values from all the syntheses, and the RMSE to their respective control trajectories was computed and is shown in Table 1 (bottom). A Kruskal-Wallis rank-sum test paired with a post-hoc Dunn test shows that distributions for each method are significantly different for both F_1 and F_2 ($p < 1e^{-3}$). Although formant modification accuracy is lower for HFC than the other methods, the RMSE difference remains below 0.1 octave in all cases, demonstrating that HFC can modify formants **using precisely the same architecture** as for modifying F_0. Subjective evaluations of F_1 and F_2 modification are left as future work, but informal tests confirm the output quality is high. Audio samples from all systems described in this paper are available online[1].

6. Conclusion

We have presented a novel approach for explicit control of an arbitrary speech parameter and shown that the system can generate high quality and accurate output for two very different categories of control parameter, one related to the glottal source and the other the vocal tract shape. Future work includes subjective evaluation of F_1/F_2 modification, as well as testing the modification of new parameters to further demonstrate the universality of our approach.

[1] Audio samples: `http://homepages.inf.ed.ac.uk/s1116548/interspeech-2020`

7. References

[1] Z. Kons, S. Shechtman, A. Sorin, C. Rabinovitz, and R. Hoory, "High quality, lightweight and adaptable TTS using LPCNet," in *Proc. of Interspeech*, Graz, Austria, September 15-19 2019, pp. 176–180.

[2] R. S. Doddipatla, N. Braunschweiler, and R. Maia, "Speaker adaptation in DNN-based speech synthesis using d-vectors," in *Proc. Interspeech*, Stockholm, Sweden, August 2017, pp. 3404–3408.

[3] Y. Wang, D. Stanton, Y. Zhang, R. J. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *Proceedings of the International Conference on Machine Learning*, vol. 80, Stockholmsmässan, Stockholm Sweden, July 10-15 2018, pp. 5180–5189.

[4] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, P. Nguyen, R. Pang, I. L. Moreno, Y. Wu *et al.*, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Advances in neural information processing systems*, 2018, pp. 4480–4490.

[5] Y. Wu, T. Hayashi, P. L. Tobing, K. Kobayashi, and T. Toda, "Quasi-periodic wavenet vocoder: A pitch dependent dilated convolution model for parametric speech generation," in *Proc. of Interspeech*, G. Kubin and Z. Kacic, Eds., September 15-19 2019, pp. 196–200.

[6] J. Valin and J. Skoglund, "LPCNet: Improving neural speech synthesis through linear prediction," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, May 2019, pp. 5891–5895.

[7] P. Govalkar, J. Fischer, F. Zalkow, and C. Dittmar, "A comparison of recent neural vocoders for speech signal reconstruction," in *ISCA Speech Synthesis Workshop*, Vienna, Austria, September 20-22 2019, pp. 7–12.

[8] J. Lorenzo-Trueba, T. Drugman, J. Latorre, T. Merritt, B. Putrycz, R. Barra-Chicote, A. Moinet, and V. Aggarwal, "Towards achieving robust universal neural vocoding," in *Proc. of Interspeech*, Graz, Austria, September 2019, pp. 181–185.

[9] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. van den Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80. Stockholmsmässan, Stockholm Sweden: PMLR, 10–15 Jul 2018, pp. 2410–2419. [Online]. Available: http://proceedings.mlr.press/v80/kalchbrenner18a.html

[10] H. Banno, H. Hata, M. Morise, T. Takahashi, T. Irino, and H. Kawahara, "Implementation of realtime STRAIGHT speech manipulation system: Report on its first implementation," *Acoustical Science and Technology*, vol. 28, no. 3, pp. 140–146, 2007.

[11] 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. E99.D, no. 7, pp. 1877–1884, 2016.

[12] M. Morise, "Cheaptrick, a spectral envelope estimator for high-quality speech synthesis," *Speech Communication*, vol. 67, pp. 1 – 7, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167639314000697

[13] O. Perrotin and I. V. McLoughlin, "A spectral glottal flow model for source-filter separation of speech," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Brighton, UK, May 12-17 2019, pp. 7160–7164.

[14] O. Perrotin and I. McLoughlin, "Gfm-voc: A real-time voice quality modification system," in *Proceedings of Interspeech*, Graz, Austria, September 15-19 2019, pp. 3685–3686.

[15] C. Hamon, E. Mouline, and F. Charpentier, "A diphone synthesis system based on time-domain prosodic modifications of speech," in *IEEE ICASSP*, 1989, pp. 238–241 vol.1.

[16] J. L. Flanagan and R. M. Golden, "Phase vocoder," *The Bell System Technical Journal*, vol. 45, no. 9, pp. 1493–1509, November 1966.

[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[18] J. Webber, "Controllable recurrent adversarial speech processing," Master's thesis, The University of Edinburgh, UK, 2019.

[19] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm, "Mutual information neural estimation," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, Stockholmsmässan, Stockholm Sweden, July 2018, pp. 531–540. [Online]. Available: http://proceedings.mlr.press/v80/belghazi18a.html

[20] K. Ito, "The LJ Speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[22] B. McFee, C. Raffel, D. Liang, D. Ellis, M. Mcvicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, January 2015, pp. 18–24.

[23] M. Evrard, S. Delalez, C. d'Alessandro, and A. Rilliard, "Comparison of chironomic stylization versus statistical modeling of prosody for expressive speech synthesis," in *Proceedings of Interspeech*, Dresden, Germany, September 6-10 2015, pp. 3370–3374.

[24] Fatchord, "Wavernn implementation," 2020. [Online]. Available: http://github.com/fatchord/WaveRNN

[25] J. Makhoul, "Linear predicion: A tutorial review," *Proceedings of the IEEE*, vol. 63, no. 4, pp. 561–580, April 1975.

[26] P. Boersma and D. Weenink, "Praat, a system for doing phonetics by computer," *Glot International*, vol. 5, no. 9/10, pp. 341–347, November 2001.