# Semi-supervised learning of glottal pulse positions in a neural analysis-synthesis framework

Frederik Bous
*UMR 9912 STMS*
*IRCAM, Sorbonne University, CNRS*
Paris, France
frederik.bous@ircam.fr

Luc Ardaillon
*UMR 9912 STMS*
*IRCAM, Sorbonne University, CNRS*
Paris, France
luc.ardaillon@ircam.fr

Axel Roebel
*UMR 9912 STMS*
*IRCAM, Sorbonne University, CNRS*
Paris, France
axel.roebel@ircam.fr

***Abstract*— This article investigates into recently emerging approaches that use deep neural networks for the estimation of glottal closure instants (GCI). We build upon our previous approach that used synthetic speech exclusively to create perfectly annotated training data and that had been shown to compare favourably with other training approaches using electroglottograph (EGG) signals. Here we introduce a semi-supervised training strategy that allows refining the estimator by means of an analysis-synthesis setup using real speech signals, for which GCI ground truth does not exist. Evaluation of the analyser is performed by means of comparing the GCI extracted from the glottal flow signal generated by the analyser with the GCI extracted from EGG on the CMU arctic dataset, where EGG signals were recorded in addition to speech. We observe that (1.) the artificial increase of the diversity of pulse shapes that has been used in our previous construction of the synthetic database is beneficial, (2.) training the GCI network in the analysis-synthesis setup allows achieving a very significant improvement of the GCI analyser, (3.) additional regularisation strategies allow improving the final analysis network when trained in the analysis-synthesis setup.**

***Index Terms*—Glottal closure instance detection, speech analysis**

## I. Introduction

Glottal Closure Instants (GCI) detection consists in finding the temporal locations of significant excitation of the vocal tract that occur in voiced speech during the closure of the vocal folds. GCI detection has many applications in speech analysis and processing [1], among which analysis of vocal disorders [2], [3], formants estimation [4], or speech synthesis [5].

One possible way to extract the GCI positions in a speech signal is to use parallel electroglottographic (EGG) recordings using dedicated hardware that measures the vocal folds contact area. GCIs can then be extracted using peak-picking on the derivative of the EGG signal. However, such recordings are rarely available, motivating the development of methods that can extract GCIs directly from speech signals. Many algorithms have been proposed for this purpose. Until recently, most approaches used to be based on hand-crafted digital signal processing techniques and heuristics. Typically, such methods first compute an intermediate speech representation, such as the linear prediction residual [6], [7], a zero-frequency filtered signal [8], or a smoothed signal [9], which emphasises the locations of glottal closure instants found at local maxima, impulses or discontinuities. Then, dynamic-programming or peak-picking is used to select the GCIs among the detected candidates.

Although such approaches have been shown to perform reasonably well, they sometimes rely on manual parameter tuning (e. g., the mean $f_0$ for SEDREAMS [9]) and the quality of their results remains quite dependant on the characteristics of the analysed speech signal (e. g., pitch and voice quality, speech or singing voice) [10]. Moreover, some algorithms like SEDREAMS or DYPSA [6] also detect GCIs during unvoiced segments and thus rely on further algorithms to filter out GCI candidates in unvoiced parts.

To overcome the limits of those methods, new data-driven approaches have been recently proposed. In [11], authors used a classification approach where GCI candidates are the negative peaks of a low-pass filtered signal. Similarly, [2] also employed a classification-based approach using 3 parallel CNNs operating on different signal representations. In [12], the authors used a CNN to optimise both a classification and regression cost, where a GCI is simultaneously detected and localised in a frame. In other related works [13], [14], the authors used neural networks to perform a regression from the speech waveform to the corresponding EGG signal using adversarial training procedures.

All these approaches rely on EGG signals for training the networks, which has two main drawbacks: (1.) the EGG signals are often noisy, and the extracted ground-truth GCIs are thus likely to contain errors; (2.) EGG signals are rarely available, which makes it difficult to build large multi-speaker databases for training, and thus limits the generalisation ability of the models.

Recently, we proposed to use high quality synthetic speech signals with almost perfect annotation for training a neural network to predict the glottal flow signal from raw speech [15]. In these initial experiments we have shown that despite being trained on synthetic speech, the resulting network compares favourably with other methods trained on EGG data when evaluated on real speech. While training with synthetic signals allows creating nearly unlimited training data with reliable an-

This work has been funded partly by the ANR project ARS (ANR-19-CE38-0001-01)

notation covering many different pulse shapes, it will not cover the full variability (pulse forms, jitter, shimmer) encountered in real speech. Notably, rough or creaky voices with irregular pulses are difficult to synthesise realistically.

To this end the following paper introduces a new semi-supervised approach that allows extending the synthetic dataset with real speech signals that do not have any associated EGG signal or GCI annotation. This is achieved by means of a complete analysis-synthesis framework that uses the detected pulses for resynthesising the original speech signal. Our evaluation shows that this analysis-synthesis setup allows significantly improving the performance on real speech.

In Section II we will present an overview of the proposed approach, network architecture, databases, and training procedure. The methodology and results of our evaluations will be presented in Section III. Finally, we will draw a conclusion in Section IV

## II. Proposed Approach

### A. The Synthetic Data Base

The power of deep learning depends on the available training data. While there exist numerous speech databases with speech, only few of these databases also contain EGG signals. Therefore GCI extraction methods that rely on the presence of EGG signals cannot fully exploit the benefits of deep learning based methods.

In previous work [15], we proposed to use a synthetic database in order to have glottal flow signals available as ground truth. The dataset consists of the publicly-available BREF [16] and TIMIT [17] datasets, that were analysed and re-synthesised using the PaN vocoder [18], [19]. The PaN vocoder uses the LF glottal source model [20], [21] as a source signal. For the resynthesis the $f_0$ estimation of [22] was used to generate the source signal by creating pulses which are separated by the fundamental period. Thus an almost perfect ground truth glottal pulse can be generated from the vocoder model.

By means of gathering the speech recordings available from previous projects we increased the size of the dataset from a few hours to about 165 hours of human voice including normal speech, singing, whispering and shouting. We call our new dataset *Agamemnon*. The simplified LF-model described in [21], which is used by the PaN vocoder, uses a single parameter to describe the pulse, the $R_d$ parameter. In [15] we resynthesised the database multiple times using different $R_d$-values in order to create a large variety of different pulse shapes. Assuming that the increased database size would provide a sufficiently large variety of $R_d$ values, only one resynthesis of the database using the $R_d$-values provided by the $R_d$ extraction algorithm [23], [24] was used in this work.

The training database thus contains a resynthesis of all speech signals produced by the PaN synthesiser annotated with all PaN parameters (GCI, $R_d$, $f_0$, voiced/unvoiced segments, spectral envelops and unvoiced components (see below)), and all real speech signals for which a restricted set of annotations will be used ($f_0$, voiced/unvoiced segments, $R_d$ analysis, spectral envelops).

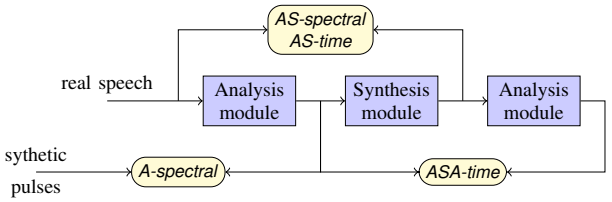


Fig. 1. Schematic of the analysis-synthesis framework and a visualisation of Table I. Neural networks are in blue, losses are in yellow. The synthesiser has as additional input a noise source signal and the spectral envelopes of the audio and of the unvoiced signal component. The two analysis modules share the same weights. The *A-time* loss is calculated separately on the synthetic speech (not included in this figure).

### B. The Analysis-Synthesis Framework

To allow the glottal flow extraction module to model the various phenomena from real speech that are not captured by the PaN resynthesis, we train it in an analysis-synthesis framework. In this framework, an analyser (the glottal flow extraction module) and a synthesiser (see below) are cascaded and trained simultaneously.

Following the approach from [15] the modules of this paper also work on audio given at a sample rate of 16kHz. Due to pooling layers in the analyser, its output sample rate is 2kHz. To match the required input sample rate of the synthesiser, upsampling using linear interpolation is performed during training. When inferring the GCI, upsampling is performed using cubic splines in order to increase accuracy.

*Synthesiser:* We use a convolutional neural network with dilated convolution based on WaveNet [25] to transform the glottal pulses generated by the analysis module, and an additional white-noise source, to raw audio. Additionally, we provide as control parameters, the spectral envelopes of the speech signal, represented in 64 mel-bands, as well as the spectral envelopes of the unvoiced speech component, represented in 16 mel-bands. The unvoiced component is extracted following the remixing approach described in [18]; the spectral envelopes are calculated according to [26].

Unlike the original WaveNet, and similar to [27] and parallel WaveNet [28] we create whole frames in one forward pass. Avoiding the recursive structure of the original WaveNet is possible due to the different objective: WaveNet requires recursion to create periodic signals. Our synthesiser has the periodic pulses as input and thus performs not much more than filtering the input according to the spectral envelopes. Furthermore, since the pulses are coherent with the speech signal, the ambiguity in the phase is minimised for the voiced speech component. Consequently, rather than learning a general distribution of raw audio speech signals under their control parameters, here we can use the speech signals as *ground truth* and require that the actual audio at hand is properly reproduced according to appropriate loss metrics (see

TABLE I
THE LOSS FUNCTION OF THE ANALYSIS-SYNTHESIS FRAMEWORK IS SPLIT INTO FIVE SEPARATE LOSSES THAT ARE ALL CALCULATED ON DIFFERENT "OUTPUTS" OF THE FRAMEWORK, EACH WITH ITS OWN TARGET VALUE. THE DETAILS FOR EACH COMPONENT ARE GIVEN BELOW.

| loss name | input 1 (network output) | input 2 (target reference) | weight | loss function | window sizes in ms |
|---|---|---|---|---|---|
| *AS-spectral* | real audio $\rightarrow$ ana $\rightarrow$ synth | real audio | 0.1 | MAE | 160, 53.3, 32.0, 22.9, 17.8 |
| *AS-time* | real audio $\rightarrow$ ana $\rightarrow$ synth (noise $= 0$) | real audio | 10 | MAE | point wise (time-domain) |
| *ASA-time* | real audio $\rightarrow$ ana $\rightarrow$ synth $\rightarrow$ ana | real audio $\rightarrow$ ana | 1 | MSE | point wise (time-domain) |
| *A-spectral* | real audio $\rightarrow$ ana | synthetic (PaN) pulses | 0.02 | MAE | 80, 40 |
| *A-time* | synthetic (PaN) audio $\rightarrow$ ana | synthetic (PaN) pulses | 10 | MSE | point wise (time-domain) |

below), which sufficiently models the speech properties when the objective is to infer the pulse positions.

Since the synthesis task is much simpler than for the original WaveNet, even simpler than in [29], we follow the observation from [29] and use a reduced size with only two stacks of six layers each (rather than three stacks of ten layers each).

### C. Loss functions

The following section describes the different loss functions that are used to train the networks. Figure 1 gives a schematic overview of the various losses used to update the networks and the flow of the signals used to calculate them.

To train the analysis-synthesis loop, we follow [27], using multi-resolution spectral loss for the resynthesis, i. e., the mean absolute error (MAE) between multiple STFT with different window lengths, which we obtain by multiplying the output segment with Hann windows of given sizes, (see Table I), applying an FFT to each of the windowed segments and taking the logarithm of the absolute values. The overlap is $1/2$ for each resolution. We call this loss *AS-spectral*.

The STFT considers only absolute values of each frequency bin. Phase differences, therefore, do not affect the spectral loss. This property is required to properly model the noise component of the signal where the phases are random. On the other hand, the pulse positions are encoded in the phases as well. The spectral loss is thus not suitable to enforce proper pulse positions.

Since the analysis module aims to provide a coherent representation for the excitation signal of both the real and the resynthesised speech signal, there is no ambiguity in phase. We can thus also apply a loss in the time domain between the real audio and the resynthesis to enforce proper reconstruction of the phases, which shall be referred to as *AS-time*. The time domain loss however will not correctly deal with noise. Therefore, when calculating the loss in the time domain, the noise input of the synthesiser is set to zero, enforcing the synthesiser to only produce the deterministic waveform.

When training the GCI analyser in the analysis-synthesis setup, the analyser and synthesiser networks need to be constrained such that they focus on their respective tasks. Notably, the analyser should not start to represent more than the glottal pulse form in its output even if such a strategy clearly would help to improve the synthesis. For imposing these constraints we rely on two strategies: In each training step, the analyser is also trained on the synthetic data, just like in [15], creating the *A-time* loss, and additionally, we include the following regularisation losses.

*a) ASA-time:* In the analysis-synthesis setup it is desired that the original signal and the resynthesised signal have the same glottal pulses. Thus we reanalyse the resynthesised audio and apply a loss in the time domain between the reanalysis and the original analysis.

Note that a global optimum for this loss is reached if the analyser does not produce any pulses. In that case all pulse signals are zero and as a result the loss is trivially zero. This behaviour was observed when the synthesiser was not pre-trained and no other regularisation was performed on the analysis of the real audio.

*b) A-spectral:* While we do not know the exact pulse positions in the real audio, we know where pulses are present (the voiced segments), the approximate pulse shape (from the $R_d$-analysis), and the pulse frequency (from the $f_0$-analysis). All these parameters are also in the synthetic glottal pulse signal. Therefore the spectrograms should be similar for the two variants of each speech signal, that are the synthetic and the real version. To enforce this invariant, we apply a spectral loss between the synthetic glottal pulse signal and the analysed glottal pulse signal. This loss is considered a major regularisation of the analyser avoiding the network to create two distinct behaviours for real and synthetic speech.

*Total loss:* The total loss is obtained as a weighted sum of all the losses. The weights were chosen manually such that the individual losses were in the same order of magnitude. The weight values can be seen in Table III.

### D. Training procedure

*Step 1 – Initialisation:* Both the analysis and the synthesis modules are trained separately on the synthetic database. For the analysis module we use the same hyper-parameters as in [15], that is, a batch size of $128$, an epoch length of $500$ updates, an initial learning rate of $2 \cdot 10^{-4}$ and decrease the learning rate by a factor of $0.75$ if no improvement on the validation loss was observed for 10 epochs. The synthesiser is trained with a batch size of $8$ training-samples from $8$ different files and a length of $2560$ audio-samples, an epoch length of $512$ updates, an initial learning rate of $5 \cdot 10^{-5}$ and the same learning rate schedule as for the analyser with a minimum of $1 \cdot 10^{-6}$. After pre-training, the batch-normalisation layers in the analyser are frozen.

*Step 2 – Analysis-Synthesis:* The analysis and synthesis modules are combined in the analysis-synthesis setup. Both real and synthetic data are used to generate the different losses. For details see Table I and Figure 1. Each training step

TABLE II
RESULTS ON CMU WITH METRICS CONSIDERING ALL GCI

| model | IDR (in %) | MR (in %) | FAR (in %) | IDA (in ms) |
|---|---|---|---|---|
| Anasynth-A | 94.21 | 1.66 | 9.71 | 0.51 |
| Anasynth-B | 92.55 | 0.87 | 15.21 | 0.64 |
| FCN-Agamemnon | 91.90 | 0.88 | 22.33 | 0.75 |
| FCN-baseline [15] | 92.73 | 2.86 | 10.87 | 0.47 |
| SEDREAMS [9] | 90.74 | 0.19 | 61.38 | 0.30 |

consists of two updates. First the losses involving real data (*AS-spectral*, *AS-time*, *ASA-time*, *A-spectral*) are calculated according to Table I and the gradients are propagated to both analyser and synthesiser. Then a "conventional" training step (as in Step I) is performed for the analyser on synthetic data to ensure that the output of the analyser does not diverge too far from the LF-model.

For the analysis-synthesis we use a batch size of $8$ with a length of $3553$ audio samples ($2560$ remain as input to the synthesiser due to "valid"-padding in the analyser), an initial learning rate of $1 \cdot 10^{-5}$ with the usual schedule (cf. Step 1) with no minimal learning rate.

## III. EVALUATION

We evaluate the performance of the analyser on the CMU arctic [30] databases using the metrics from [15], which are based on the metrics defined in [31]. The results are summarised in Table II. The GCI are obtained using a simple peak-picking procedure on the negative peaks of the derivative of the predicted glottal flow, as was done in [15].

### A. Test Setup

We compare the following configurations:

- **SEDREAMS** For comparison, we include the results from SEDREAMS [9] as a baseline.
- **FCN-baseline** As another baseline we use the configuration "FCN-synth-GF" from [15]. This is assumed to be the current state-of-the-art.
- **FCN-Agamemnon** The same network and training procedure as FCN-baseline but trained on the synthetic (not augmented) part of the *new*, larger database (Agamemnon).
- **Anasynth-A** The analysis module from the analysis-synthesis as introduced in the previous section. It was initialised with FCN-Agamemnon.
- **Anasynth-B** The analysis module from the analysis-synthesis, however trained without *A-spectral*. Also initialised with FCN-Agamemnon.

### B. Evaluation Metrics

We use the same evaluation metrics that were used for Table 2 in [15], which are defined in Table III. For each model we calculate the metrics for each file separately and create an average for each of the speakers, *bdl*, *jmk* and *slt*. We then create total averages, as the average over each speaker, i. e., equally representing each *speaker*, and display the result in Table II.

TABLE III
EVALUATION METRICS USED IN TABLE II

| | |
|---|---|
| IDR | (*identification rate*) The number of unique identifications between generated pulses and pulses in the ground truth divided by the total number of pulses in the ground truth. |
| MR | (*miss rate*) The number of pulses in the ground truth that were not detected by the system divided by the total number of pulses in the ground truth. |
| FAR | (*false alarm rate*) The number of generated pulses that could not be associated with any pulse from the ground truth divided by the total number of pulses in the ground truth. |
| IDA | (*identification accuracy*) Standard deviation of the misalignment between the associated detected and ground truth GCI. |

### C. Observations

*1) Having a large variety of pulse shapes in the synthetic data is important:* Although the database for FCN-Agamemnon was much larger than the database for FCN-baseline, the identification rate of FCN-Agamemnon is slightly worse than the one of FCN-baseline. This can be explained by the larger variety of $R_d$ values in the synthetic database for FCN-baseline obtained by the data augmentation (multiple resynthesis with different $R_d$-values).

It is worth noting that although this paper focuses on estimating the glottal closure instances (GCI), the analyser learns much more than just the GCI: It learns the whole pulse shape, that is, in the context of the underlying pulse model (LF-model [21]) it learns the proper $R_d$-value. Thus augmenting the variety in $R_d$-values/pulse-shapes improves generalisation. Furthermore, the pulses generated by the analysis module could potentially be used to infer the $R_d$-value, which might improve the $R_d$-analysis and consequently the resynthesis of the synthetic database.

*2) Refining the analysis as part of the analysis-synthesis framework does improve the GCI detection:* We see that the identification rate of Anasynth-A surpasses its initialisation FCN-Agamemnon and both baselines. We can conclude that refining the analysis in the analysis-synthesis framework does positively impact the performance of the analysis module.

This can be explained by the fact that the refinement happens on real data and the real data is closer to the evaluation dataset, CMU arctic, than the synthetic data and suggests that the analysis-synthesis setup is able to force the analysis module to generate an expressive pulse model. Consequently the model generalises better to real data,

Observe that none of the losses used for training explicitly represents the GCI. Nevertheless our model is able to detect the GCI better than any other method. This suggests that the model does not only learn the pulse positions, it really learns the pulse shapes from which the GCI can be derived.

*3) Regularisation of the pulse signal is crucial:* The improvement of Anasynth-B is smaller than for Anasynth-A compared to FCN-Agamemnon and still does not allow to surpass the results of FCN-baseline. This shows that the regularisation in terms of the *A-spectral* is a crucial element of this setup. One problem with the analysis-synthesis method is the fact that the synthesiser learns only to use the pulse input in

voiced speech. Consequently the predicted pulse signal has no effect at the synthesiser output during unvoiced speech and silence and thus does not affect the *AS* losses. Since most of the error happens in the unvoiced segments, the improvement in configuration Anasynth-B is minimal.

In an additional experiment it was shown that training the analyser on real data with *A-spectral* and on synthetic data with *A-time*, hence avoiding the use of a synthesiser, does not allow improving over the FCN-Agamemnon baseline.

## IV. Conclusion

In order to further improve our previous results [15], we proposed a new semi-supervised approach for training a neural GCI detector that allows training with controlled synthetic speech and real speech data. The main idea of this approach is to incorporate our previously-proposed neural network for predicting the glottal flow into a constrained analysis-synthesis loop trained using a reconstruction loss and a collection of additional loss functions that ensure proper segregation of the analysis and synthesis tasks. To better represent all the variability of voice signals, we also proposed to use here a much larger database with many different types of signals.

For the task of GCI detection, evaluation on the CMU arctic database has shown that the refinement step using real data significantly improves the identification rate of the estimator with respect to training it solely on the synthetic database. The identification rate also surpasses all previous systems establishing the proposed method as the new state-of-the-art. While we only assessed in this paper the performance of our analyser for GCI detection, the predicted glottal flow may also be used for other tasks like $R_d$ analysis.

## References


[1] T. Drugman, P. Alku, A. Alwan, and B. Yegnanarayana, "Glottal source processing: From analysis to applications," *Computer Speech and Language*, vol. 28, no. 5, pp. 1117–1138, 2014.

[2] M. G. Reddy, T. Mandal, and K. S. Rao, "Glottal closure instants detection from pathological acoustic speech signal using deep learning," in *arXiv*, 2018.

[3] P. S. Deshpande and M. S. Manikandan, "Effective glottal instant detection and electroglottographic parameter extraction for automated voice pathology assessment," *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 2, pp. 398–408, 2018.

[4] J. M. Anand, S. Guruprasad, and B. Yegnanarayana, "Extracting formants from short segments of speech using group delay functions," in *INTERSPEECH*, 2006, pp. 1009–1012.

[5] T. Drugman and T. Dutoit, "The Deterministic plus Stochastic model of the residual signal and its applications," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 3, pp. 968–981, 2012.

[6] P. A. Naylor, A. Kounoudes, J. Gudnason, and M. Brookes, "Estimation of glottal closure instants in voiced speech using the DYPSA algorithm," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 1, pp. 34–43, 2007.

[7] A. P. Prathosh, T. V. Ananthapadmanabha, and A. G. Ramakrishnan, "Epoch extraction based on integrated linear prediction residual using plosion index," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 12, pp. 2471–2480, 2013.

[8] K. S. R. Murty and B. Yegnanarayana, "Epoch extraction from speech signals," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 8, pp. 1602–1613, 2008.

[9] T. Drugman and T. Dutoit, "Glottal closure and opening instant detection from speech signals," in *INTERSPEECH*, 2009.

[10] O. Babacan, T. Drugman, N. D'Alessandro, N. Henrich, and T. Dutoit, "A quantitative comparison of glottal closure instant estimation algorithms on a large variety of singing sounds," in *INTERSPEECH*, 2013, pp. 1702–1706.

[11] S. Yang, Z. Wu, B. Shen, and H. Meng, "Detection of glottal closure instants from speech signals: A convolutional neural network based method," in *INTERSPEECH*, 2018, pp. 317–321.

[12] M. Goyal, V. Srivastava, and A. P. Prathosh, "Detection of glottal closure instants from raw speech using convolutional neural networks," in *INTERSPEECH*, 2019.

[13] A. P. Prathosh, V. Srivastava, and M. Mishra, "Adversarial approximate inference for speech to electroglottograph conversion," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 2183–2196, 2019.

[14] K. Deepak, P. Kulkarni, U. Mudenagudi, and S. Prasanna, "Glottal instants extraction from speech signal using generative adversarial network," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5946–5950.

[15] L. Ardaillon and A. Roebel, "Gci detection from raw speech using a fully-convolutional network," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, accept. for pub.

[16] J. L. Gauvain, L. F. Lamel, and M. Eskenazi, "Design considerations and text selection for BREF, a large French read-speech corpus," in *ICSLP*, 1990, pp. 1097–1100.

[17] V. Zue, S. Seneff, and J. Glass, "Speech database development at MIT: TIMIT and beyond," *Speech communication*, vol. 9, no. 4, pp. 351–356, 1990.

[18] S. Huber and A. Roebel, "On glottal source shape parameter transformation using a novel deterministic and stochastic speech analysis and synthesis system," in *Proc InterSpeech*, 2015.

[19] L. Ardaillon, "Synthesis and expressive transformation of singing voice," Ph.D. dissertation, EDITE; UPMC-Paris 6 Sorbonne Universités, 2017.

[20] G. Fant, J. Liljencrants, and Q.-g. Lin, "A four-parameter model of glottal flow," *STL-QPSR*, vol. 4, no. 1985, pp. 1–13, 1985.

[21] G. Fant, "The lf-model revisited. transformations and frequency domain analysis," *Speech Trans. Lab. Q. Rep., Royal Inst. of Tech. Stockholm*, vol. 2, no. 3, p. 40, 1995.

[22] L. Ardaillon and A. Roebel, "Fully-convolutional network for pitch estimation of speech signals," *Proc. Interspeech 2019*, pp. 2005–2009, 2019.

[23] G. Degottex, A. Roebel, and X. Rodet, "Phase minimization for glottal model estimation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 5, pp. 1080–1090, 2010.

[24] S. Huber, A. Roebel, and G. Degottex, "Glottal source shape parameter estimation using phase minimization variants," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[25] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio." in *SSW*, 2016, p. 125.

[26] A. Röbel, F. Villavicencio, and X. Rodet, "On cepstral and all-pole based spectral envelope modeling with unknown model order," *Pattern Recognition Letters, Special issue on Advances in Pattern Recognition for Speech and Audio Processing*, vol. 28, no. 6, pp. 1343–1350, 2007.

[27] X. Wang, S. Takaki, and J. Yamagishi, "Neural source-filter waveform models for statistical parametric speech synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 402–415, 2019.

[28] A. v. d. Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. v. d. Driessche, E. Lockhart, L. C. Cobo, F. Stimberg *et al.*, "Parallel wavenet: Fast high-fidelity speech synthesis," *arXiv preprint arXiv:1711.10433*, 2017.

[29] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," *arXiv preprint arXiv:1712.05884*, 2017.

[30] J. Kominek and A. W. Black, "The cmu arctic speech databases," in *Fifth ISCA workshop on speech synthesis*, 2004.

[31] M. R. Thomas and P. A. Naylor, "The sigma algorithm: A glottal activity detector for electroglottographic signals," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 8, pp. 1557–1566, 2009.