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

Nonlinear Residual Echo Suppression using a Recurrent Neural Network

Lukas Pfeifenberger, Franz Pernkopf

Signal Processing and Speech Communication Laboratory
Graz University of Technology, Graz, Austria

lukas.pfeifenberger@alumni.tugraz.at
pernkopf@tugraz.at

Abstract

The acoustic front-end of hands-free communication devices introduces a variety of distortions to the linear echo path between the loudspeaker and the microphone. While the amplifiers may introduce a memory-less non-linearity, mechanical vibrations transmitted from the loudspeaker to the microphone via the housing of the device introduce non-linarities with memory, which are much harder to compensate. These distortions significantly limit the performance of linear Acoustic Echo Cancellation (AEC) algorithms. While there already exists a wide range of Residual Echo Suppressor (RES) techniques for individual use cases, our contribution specifically aims at a low-resource implementation that is also real-time capable. The proposed approach is based on a small Recurrent Neural Network (RNN) which adds memory to the residual echo suppressor, enabling it to compensate both types of non-linear distortions. We evaluate the performance of our system in terms of Echo Return Loss Enhancement (ERLE), Signal to Distortion Ratio (SDR) and Word Error Rate (WER), obtained during realistic double-talk situations. Further, we compare the postfilter against a state-of-the art implementation. Finally, we analyze the numerical complexity of the overall system.

Index Terms: Acoustic echo cancellation, residual echo suppression, non-linear echo, recurrent neural networks

1. Introduction

In hands-free speech communication devices, an *Acoustic Echo Canceler* (AEC) is an essential building block which models the acoustic path between loudspeaker output and microphone input with a linear *Finite Impulse Response* (FIR) filter. The AEC subtracts the echo replica from the microphone signal, enabling echo-free voice communication [1]. Unfortunately, he task of echo cancellation is complicated by additional non-linear distortions in the loudspeaker and the amplifier, and also by mechanical vibrations transmitted from the loudspeaker via the case of the device to the microphone [2]. These distortions cannot be modeled by linear echo cancelers. Consequently, the practically achievable Echo Return Loss Enhancement (ERLE) is limited, which results in a degraded speech quality and intelligibility. This problem is even more relevant today as speakerphones or smart speakers are portable devices with small enclosure dimensions and tiny loudspeakers, which are prone to non-linear distortions. Despite their size, they produce high sound pressure levels by using amplifiers which pre-distort the loudspeaker signal [3]. This introduces even more distortions to the echo path. Non-linear distortions can be categorized into two groups:

1) Non-linearities without memory, i.e. harmonic distortions caused by non-linear loudspeaker drivers, or clipping of the microphone signal [4]. Non-linear systems without memory can be approximated by polynomials in the form $f_{NL}(x) = \sum_{i=0}^{\infty} \alpha_i \cdot x^i$. The parameters α_i may be determined using non-linear system identification, i.e. by using a chirp signal [5]. This is also a standard procedure for measuring the individual harmonics and the overall *Total Harmonic Distortions* (THD) of loudspeakers and amplifiers. Harmonic distortions may be compensated by incorporating *power-filters* into the AEC algorithm [6–8], or by using a residual echo suppressor [9–12].

2) Non-linearities with memory, i.e. partial vibrations of the loudspeaker membrane, or structure-borne sounds and mechanical vibrations [4]. Non-linear systems with memory can be approximated by Volterra series [13]. As the size of a Volterra kernels grows exponentially with its order, this concept is of limited use in real-world applications. Further, tuning the kernels for a given non-linearity is a non-trivial task, as system identification requires *Higher Order Statistics* (HOS) and *spectral analysis*. However, several echo suppressors with Volterra series have been proposed, e.g.: sparse Volterra kernels [14, 15], or Hammerstein models [16, 17].

Both Volterra series and *Multilayer Perceptrons* (MLP) are universal approximators for non-linearities with memory. Consequently, neural networks have been proposed for non-linear residual echo cancellation [18–23]. However, we found that many contributions in this field are limited by one or more of the following aspects: (i) Only memoryless non-linearities are considered, even though both types always occur in a real-world scenario [4]. (ii) The neural network features a lot of weights, making the postfilter computationally more expensive than the actual AEC itself. (iii) The system is not real-time capable due to the data flow of the neural network.

In this paper, we consider *Recurrent Neural Networks* (RNNs) as postfilter to address these shortcomings. (i) Due to the recurrent structure of our neural network, non-linearities with memory can be learned directly from real-world audio examples. The use of internal memory in form of an LSTM layer allows for a smaller network compared to an MLP [18, 23]. (ii) Our approach is real-time capable, due to the LSTM layer operating only in forward direction of the data stream. It introduces no additional delay to the overall system, as it operates on one block of data at a time. (iii) With only two Dense layers and one LSTM layer with 25 units in its smallest variant, our neural network is considerably smaller than comparable approaches with 1024 or more units. Further, our system can be trained with as little as 1.75h of echo recordings, which allows for a fast training process even without a GPU.

2. System Model

We assume a classical, monaural speakerphone with a loudspeaker and a microphone for hands-free telephony applications, i.e. *Voice over IP* (VoIP). The system model is shown in Fig. (1). In this setup, the far-end speaker signal is received via the network (RX), and the near-end speaker signal is transmitted (TX) back over the network. Due to acoustic echoes, the microphone picks up both the near-end speaker and the acoustic echo from the loudspeaker. Hence, an AEC is required. In Fig. (1), all signals are denoted in the *Short Time Fourier Transform* (STFT) domain with a frequency index k and a time index t. The loudspeaker and microphone signals are represented by $X(k,t)$ and $D(k,t)$, respectively. The echo model, which is obtained from the AEC filter, is given as $Y(k,t)$. By subtracting the echo model from the microphone signal, we obtain the residual signal $E(k,t)$. The proposed postfilter operates on the residual and the microphone signal, and outputs the enhanced signal $Z(k,t)$.

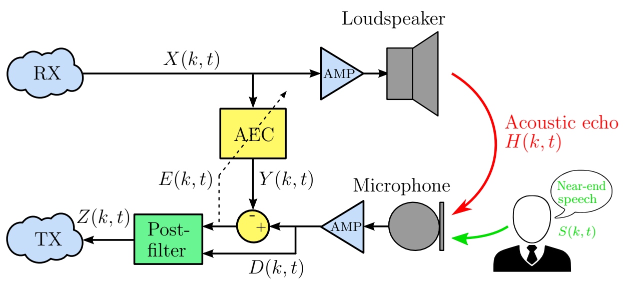

Figure 1: *System Model with signals in the STFT domain.*

The *Echo Impulse Response* (EIR) $H(k,t)$ is modeled as FIR filter. Usually, it is much longer than the STFT block length, therefore it is partitioned into L blocks. Using this notation, the microphone signal can be written as

$$D(k,t) = S(k,t) + f_{NL}\big(X(k,t)\big) + \sum_{l=t-L}^{t} X(k,l)H(k,l), \tag{1}$$

where $S(k,t)$ denotes near-end speech signal, and $f_{NL}(\cdot)$ denotes an unknown non-linear relationship with memory. The AEC in Fig. (1) estimates the EIR $\hat{H}(k,t)$, such that the echo model is given as

$$Y(k,t) = \sum_{l=t-L}^{t} X(k,l)\hat{H}(k,l) \tag{2}$$

After the subtraction stage, the residual is given as

$$E = D - Y = \sum X\tilde{H} + f_{NL}\big(X\big) + S, \tag{3}$$

where $\tilde{H} = H - \hat{H}$. The frequency and time indices have been omitted for readability. Ideally, the filter mismatch $\tilde{H}$ and the non-linearity $f_{NL}(\cdot)$ are small, so that the residual signal contains only the near-end speech signal $S(k,t)$.

2.1. AEC Framework

We use a frequency-domain, block-based Acoustic Echo Canceler (AEC), which partitions the echo filter into multiple blocks using a STFT. This reduces the overall system delay of the algorithm to a single STFT block length, allowing for real-time operation. We chose the state-space block-partitioned AEC implementation from [24], which we found to be both robust and well-performing in real-world scenarios. We use a block length of 1024 samples, 50% overlap, and $L = 16$ blocks in total at $f_s = 16$kHz to model a tail length of up to 512ms.

In a practical application there is always a mismatch between the filter estimated by the AEC, and the actual EIR. The linear echo path may change over time as the near-end speaker moves in front of the device. The device itself may be carried around, causing a constantly changing EIR. These changes must be tracked by the AEC algorithm.

3. RNN postfilter

In a real-world scenario with actual loudspeakers and amplifiers, both non-linearities with and without memory are always present. These distortions cannot be compensated by the AEC. Residual echo suppressors have been proposed for both non-linearities without memory [8–12], and for non-linearities with memory [14–17]. With the advent of machine learning, the performance of residual echo suppressors has dramatically increased [18–23].

However, our contribution differs in the following key aspects: (i) Due to the recurrent structure of our neural network, non-linearities with memory can be learned directly from real-world audio examples, while most contributions only use memoryless non-linearities. (ii) Our approach is real-time capable, due to the LSTM layer operating only in forward direction of the data stream. It introduces no additional delay to the overall system. (iii) With only three layers and 25 LSTM cells in its smallest variant, our neural network is considerably smaller than comparable approaches [18, 23].

Fig. (2) outlines the architecture of our RNN postfilter. It consists of three layers, and operates on log-differences of the power of the microphone signal $D(k,t)$ and the echo model $Y(k,t)$. The first layer is a simple dense layer, which performs data compression from K frequency bands to M bands. This is useful to facilitate a small LSTM layer, which is the second layer of the system and the computationally most complex one. M can be as low as 25 units, whereas $K = 513$ in our implementation. The third layer expands the data back to K bands. It predicts a *gain mask* $p(k,t)$, which is multiplied element-wise to the residual signal $E(k,t)$ to produce the enhanced output, i.e.:

$$Z(k,t) = E(k,t)p(k,t), \tag{4}$$

with $p(k,t) \in [0,1]$.

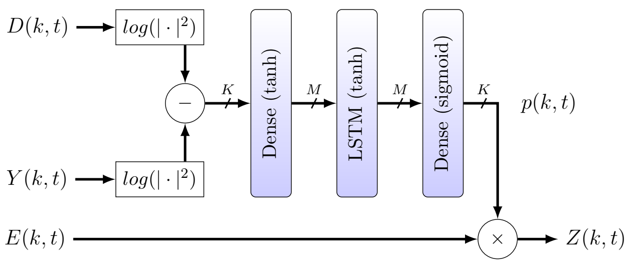

Figure 2: *RNN architecture with K frequency bins and M LSTM units.*

3.1. Hybrid loss function

To train the RNN postfilter, we consider two use cases during a conversation:

- *Single-talk*: Only the far-end speaker $X(k,t)$ is talking, the near-end is silent, i.e. $S(k,t) = 0$.
- *Double-talk*: Both near- and far-end speakers talk simultaneously.

During single-talk, we want to maximize the *Echo Return Loss Enhancement* (ERLE), i.e.: the output $Z(k,t)$ is ideally zero. The ERLE is defined as follows:

$$\mathcal{L}_{\text{ERLE}} = 10log_{10}\frac{\sum_{K,T}|D(k,t)|^2}{\sum_{K,T}|Z(k,t)|^2} \tag{5}$$

During double-talk, we want to maximize the *Signal to Distortion Ratio* (SDR), i.e.: the output $Z(k,t)$ is identical to the near-end signal $S(k,t)$. The SDR is defined as:

$$\mathcal{L}_{\text{SDR}} = 10log_{10}\frac{\sum_{K,T}|S(k,t)|^2}{\sum_{K,T}|S(k,t) - Z(k,t)|^2} \tag{6}$$

To fulfill both constraints, we use a hybrid objective to train the RNN postfilter. The overall loss function to be minimized by the RNN is given as:

$$\mathcal{L} = -\mathcal{L}_{\text{ERLE}} - \lambda\mathcal{L}_{\text{SDR}}, \tag{7}$$

where the parameter λ allows to adjust the importance of either the ERLE or SDR constraint during training.

4. Experiments

4.1. Recording Setup

In order to obtain realistic distortions which contain both types of non-linearities, it is essential to use a real-world setup, i.e. a speakerphone or smart speaker with a loudspeaker and a microphone in the same case. Otherwise it would be difficult to accurately simulate realistic non-linearities with memory, as well as changing EIR paths over time. Therefore, we used a small speakerphone (EasyAcc-MC) with a 3W loudspeaker and an electret microphone. We disconnected the internal electronics and used an external amplifier to drive the loudspeaker. The amplifier and the microphone were plugged into the line-out and mic-in jack of a sound card, respectively. We measured the *Total Harmonic Distortion* (THD) of the speakerphone at 3W, which is about 12%. Therefore, a reasonable amount of non-linear distortions is present in our setup [2]. To drive the speakerphone from a Linux-based PC with ALSA [25], we use the *PlayRec* Python module [26], which simultaneously plays and records audio from a sound card. We further implemented the block-based AEC from [24] in Python, to obtain the relevant signals for training the RNN postfilter.

The speakerphone was placed in 7 different office rooms in 10 different positions each. The rooms had a RT_{60} between 250ms and 500ms. For each position, we generated 3 training examples. Each training example consists of the excitation signal $X(k,t)$, and the recorded echo response $D(k,t)$. We use 30s of randomly concatenated utterances from the TIMIT speech corpus [27] as excitation signal $X(k,t)$. The simultaneously recorded microphone signal $D(k,t)$ contains 30s echo response. In total, 1.75 hours of reverberated samples have been obtained. All samples were recorded at f_s = 16kHz. Fig. (3) illustrates the recording setup, using the speakerphone. Green arrows represent the linear echo path (EIR), and red parts depict potential sources of non-linear distortions.

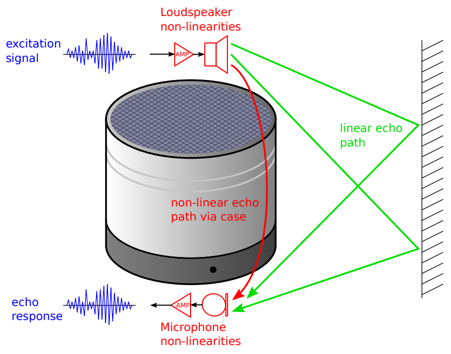
Loudspeaker non-linearities
excitation signal
AMP
linear echo path
non-linear echo path via case
echo response
AMP
Microphone non-linearities

Figure 3: *Recording setup using a small speakerphone.*

4.2. Training

We used the recordings from the first 6 rooms for training, and the rest for evaluating the RNN postfilter. Note that the neural network does not learn speech or speaker characteristics, but rather the non-linearities embedded in the microphone signal $D(k,t)$. Hence, a small training set is sufficient. We train the RNN as follows:

First, we process each of the 30s long data samples with the AEC algorithm. The AEC provides the residual $E(k,t)$ and the echo model $Y(k,t)$, which are required as inputs for the RNN (see Fig. 2). To train on time-varying EIRs, we reset the AEC weights at the beginning of each 30s long training example.

To optimize the RNN for both ERLE and SDR, we use each training example twice: In the first pass, the ERLE from Eq. (5) is calculated for the single-talk case, i.e. the near-end speaker $S(k,t) = 0$. In the second pass, the SDR from Eq. (6) is calculated for the double-talk case. We used randomly selected utterances from the the *si_tr_s* set of the WSJ0 [28] corpus to simulate the near-end speaker $S(k,t)$, which we mixed into the microphone signal with a *Signal to Echo Ratio* (SER) of $-12dB$. This corresponds to the SER encountered when driving the loudspeaker at 3W and speaking into the device from approximately 0.5m distance. The trade-off parameter λ in Eq. (7) was set to 1. We trained 7 different versions of the RNN postfilter, where we parametrized the size of the LSTM layer from 25 to 250 units, see also Table (1).

4.3. Testing

Testing the RNN postfilter was done with the unused recordings from the 7th room. ERLE and SDR are evaluated as during training. We also measured the *Word Error Rate* (WER) for the enhanced signal $Z(k,t)$ during double talk. The WER is obtained by the Google Speech-to-Text API [29]. In particular, it was measured using clean WSJ0 data set as reference, for which the Google Speech-to-Text API reports a WER of 6.1%.

4.4. Results

Table 1 reports the ERLE, SDR and WER scores for experiments using a varying LSTM layer size from M = 25 to 250 units. As a baseline, we also evaluated the AEC without the postfilter. Further, we compare our postfilter to a state-of-the art reference AEC implementation (Speex-DSP) [30]. Speex also uses a frequency-domain, block-based echo canceler [31], and a residual echo-suppressor. We configured the same echo-tail length of 512ms. It can be seen that Speex slightly outperforms the baseline in all scores. However, our RNN postfilter yields a significant improvement in all scores.

Table 1: *ERLE, SDR and WER scores for the RNN postfilter, the reference system (Speex-DSP) and the AEC without a postfilter as a basline.*

LSTM cells M	ERLE	SDR	WER
25	44.868	11.079	17.08%
50	51.802	12.084	16.41%
75	55.303	12.656	14.87%
100	60.447	12.902	12.56%
150	61.650	13.294	11.72%
200	60.637	13.404	11.33%
250	63.019	13.434	10.64%
Speex-DSP	21.726	6.716	25.16%
no postfilter	19.206	5.454	44.73%

4.5. Performance

Fig. (4) illustrates a 30s example from the test set with $M = 100$ LSTM cells. Panel (a) shows the far-end and near-end signals, respectively. Panel (b) shows the residual signal $E(k,t)$. It can be seen that the AEC needs approximately 10s to adjust to the EIR. During that time, the error in the residual is quite large. About 16s into the sample, the near-end speaker $S(k,t)$ starts talking. Panel (c) shows the enhanced output $Z(k,t)$ of the RNN postfilter. It can be seen that the enhanced signal only contains the desired speech signal. Panel (d) shows the ERLE, measured over time and split into the contribution of both the AEC and the postfilter, respectively.

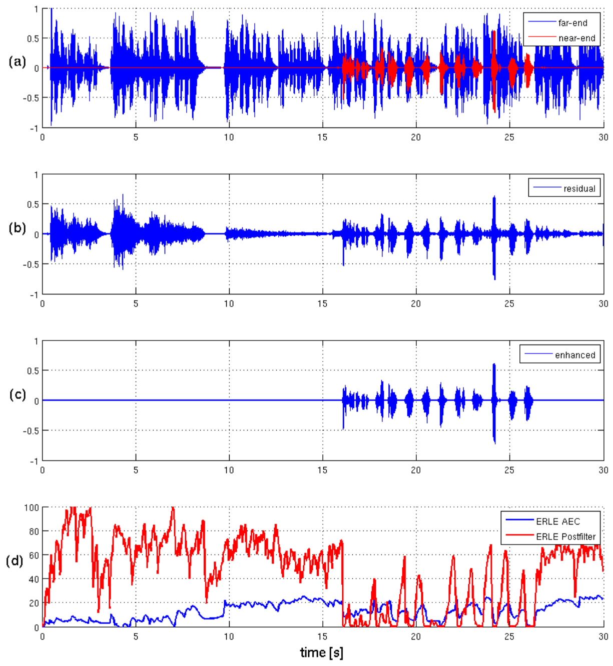

Figure 4: *Performance of the RNN postfilter for a 30s test example: (a) far-end and near-end signals $X(k,t)$ and $S(k,t)$, respectively. (b) AEC residual $E(k,t)$. (c) enhanced output of the postfilter $Z(k,t)$. (d) ERLE of the AEC and the postfilter.*

4.6. Numerical complexity

In this section we will discuss the numerical complexity of the overall system. We count the total number of *Multiply and Accumulate* (MAC) operations, which can be performed on either a dedicated DSP or CPU with a vector floating point unit (i.e.: ARM NEON). The RNN postfilter consists of 3 layers. The first layer is a dense layer with K inputs and M outputs. Its forward path is defined as $y = \boldsymbol{W}x + b$, where $\boldsymbol{W}$ is a $K \times M$ weight matrix and b is a bias vector of size M, and the input $x \in \mathbb{R}^K$. Hence, the layer requires $(K \cdot M + M)$ MAC operations[1]. In the same manner, the LSTM layer requires $(8M^2 + 7M)$ MACs, and the third layer requires $(M \cdot K + K)$ MACs.

The complexity of the state-space block-partitioned AEC can be assessed using Eq. (26-32) in [24]. For $L = 16$ blocks and $K = 513$ frequency bins, we obtain 143k MACs including complex operations. Additionally, there are $L + 3$ complex FFTs and $L + 1$ complex IFFTs required for zero-padding and processing the time-domain inputs, adding another 737k MACs to the algorithm. Table (2) summarizes the numerical complexity for each RNN postfilter and the AEC. It can be seen that the postfilter adds only a fraction to the overall complexity, especially for small LSTM layers. At $f_s = 16$kHz and a block length of 1024 samples with 50% overlap, we process 31.25 blocks per second. In total, the smallest postfilter+AEC requires 25M MACs/s, while the largest postfilter+AEC requires 51M MACs/s, which is well within the reach of modern embedded systems.

Table 2: *Numerical complexity per block.*

LSTM cells M	MAC operations
25	31k
50	72k
75	123k
100	184k
150	336k
200	527k
250	758k
AEC	880k

5. Conclusion

In this paper, we proposed a residual echo suppressor which uses a recurrent neural network to model distortions such as non-linearities with memory, which are often found in small speakerphones housing a loudspeaker and a microphone in the same case. We showed that our approach uses very little resources, while still being real-time capable as it introduces no additional delay to the echo canceler. We also showed that the performance in terms of ERLE, SDR and WER is greatly improved compared to a state-of-the art echo canceler and residual echo suppressor. In particular, the RNN postfilter lowers the WER by up to 14.52%

6. References

[1] S. Haykin, *Adaptive Filter Theory*, 4th ed. New Jersey: Prentice Hall, 2002.

[1]We excluded the numerical complexity of the *tanh* or *sigmoid* activation functions, as they are negligible compared to the matrix multiplications, and also implemented in highly optimized libraries.

[2] H. Kuttruff, *Room Acoustics*, 5th ed. London–New York: Spoon Press, 2009.

[3] Y. A. Huang and J. Benesty, *Audio Signal Processing For Next-Generation Multimedia Communication Systems*. Boston: Kluwer Academic Publishers, 2004.

[4] T. D. Rossing, *Springer Handbook of Acoustics*. Berlin–Heidelberg–New York: Springer, 2007.

[5] A. Novak, L. Simon, F. Kadlec, and P. Lotton, "Nonlinear system identification using exponential swept-sine signal," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 8, pp. 2220–2229, 2010.

[6] F. Kuech, A. Mitnacht, and W. Kellermann, "Nonlinear acoustic echo cancellation using adaptive orthogonalized power filters," in *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, vol. 3, 2005, pp. iii/105–iii/108 Vol. 3.

[7] F. Kuech and W. Kellermann, "Orthogonalized power filters for nonlinear acoustic echo cancellation," *Signal Processing*, vol. 86, no. 6, pp. 1168 – 1181, 2006, applied Speech and Audio Processing.

[8] F. Kuech and W. Kellermann, "Nonlinear residual echo suppression using a power filter model of the acoustic echo path," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, vol. 1, 2007, pp. I–73–I–76.

[9] D. A. Bendersky, J. W. Stokes, and H. S. Malvar, "Nonlinear residual acoustic echo suppression for high levels of harmonic distortion," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008, pp. 261–264.

[10] Kun Shi, Xiaoli Ma, and G. Tong Zhou, "A residual echo suppression technique for systems with nonlinear acoustic echo paths," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008, pp. 257–260.

[11] S. Malik and G. Enzner, "State-space frequency-domain adaptive filtering for nonlinear acoustic echo cancellation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 7, pp. 2065–2079, 2012.

[12] S. Malik and G. Enzner, "A variational bayesian learning approach for nonlinear acoustic echo control," *Signal Processing, IEEE Transactions on*, vol. 61, pp. 5853–5867, 12 2013.

[13] H. Enzinger, K. Freiberger, G. Kubin, and C. Vogel, "Fast time-domain volterra filtering," in *2016 50th Asilomar Conference on Signals, Systems and Computers*, 2016, pp. 225–228.

[14] A. Guerin, G. Faucon, and R. Le Bouquin-Jeannes, "Nonlinear acoustic echo cancellation based on volterra filters," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 672–683, 2003.

[15] F. Kuech and W. Kellermann, "A novel multidelay adaptive algorithm for volterra filters in diagonal coordinate representation [nonlinear acoustic echo cancellation example]," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, 2004, pp. ii–869.

[16] S. Malik and G. Enzner, "Fourier expansion of hammerstein models for nonlinear acoustic system identification," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 85–88.

[17] A. Schwarz, C. Hofmann, and W. Kellermann, "Combined nonlinear echo cancellation and residual echo suppression," in *Speech Communication; 11. ITG Symposium*, 2014, pp. 1–4.

[18] C. M. Lee, J. W. Shin, and N. S. Kim, "Dnn-based residual echo suppression," in *INTERSPEECH*, 2015.

[19] T. V. Huynh, "A new method for a nonlinear acoustic echo cancellation system," 2017.

[20] H. Zhang and D. Wang, "Deep learning for acoustic echo cancellation in noisy and double-talk scenarios," 09 2018, pp. 3239–3243.

[21] G. Carbajal, R. Serizel, E. Vincent, and E. Humbert, "Multiple-input neural network-based residual echo suppression," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 231–235.

[22] G. Carbajal, R. Serizel, E. Vincent, and E. Humbert, "Joint dnn-based multichannel reduction of acoustic echo, reverberation and noise," 2019.

[23] Q. Lei, H. Chen, J. Hou, L. Chen, and L. Dai, "Deep neural network based regression approach for acoustic echo cancellation," in *Proceedings of the 2019 4th International Conference on Multimedia Systems and Signal Processing*, ser. ICMSSP 2019. New York, NY, USA: Association for Computing Machinery, 2019, p. 94–98.

[24] F. Kuech, E. Mabande, and G. Enzner, "State-space architecture of the partitioned-block-based acoustic echo controller," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 1295–1299.

[25] "Alsa-project," Website, visited on February 19th 2020. [Online]. Available: https://alsa-project.org/wiki/Main_Page

[26] "python-sounddevice," Website, visited on February 19th 2020. [Online]. Available: https://python-sounddevice.readthedocs.io/en/0.3.15/

[27] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "Darpa timit acoustic phonetic continuous speech corpus cdrom," 1993.

[28] D. B. Paul and J. M. Baker, "The design for the wall street journal-based csr corpus," in *Proceedings of the Workshop on Speech and Natural Language*, ser. HLT '91. Stroudsburg, PA, USA: Association for Computational Linguistics, 1992, pp. 357–362.

[29] "SpeechRecognition – a library for performing speech recognition, with support for several engines and apis, online and offline." Website, 2018, visited on March 25th 2020. [Online]. Available: https://pypi.org/project/SpeechRecognition/

[30] "Speex-dsp," Website, visited on February 19th 2020. [Online]. Available: https://github.com/xiongyihui/speexdsp-python

[31] J. . Soo and K. K. Pang, "Multidelay block frequency domain adaptive filter," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 2, pp. 373–376, 1990.