We introduce a general and data-driven adaptation-control framework for stereophonic acoustic echo cancellation. The adaptation update rule for the filters that estimate the actual echo paths is compactly expressed with the widely-linear model in the complex time domain. A single step-size parameter that governs the behavior of the adaptation process is optimized by minimizing the misalignment between the actual echo paths and their filtered estimate. The relation between acoustic signals and the optimal step-size is learned via a deep neural network. In test mode, the optimal step-size prediction is inferred by the network and fed to the sign-error normalized least mean-squares (SNLMS) adaptive filter for echo-paths tracking. Real and simulated data show advantageous performance in single and double-talk scenarios across various acoustic setups.

Index Terms— Stereophonic acoustic echo cancellation, adaptation control, variable step-size, sign-error NLMS, deep learning.

1. INTRODUCTION

In stereophonic hands-free speech communication, the near-end microphones may capture three types of acoustic signals: the desired speech, additional noises, and reverberant echoes. The echoes may impede conversation intelligibility as perceived by the far-end participant. The stereophonic acoustic echo cancellation (SAEC) task is two-fold; tracking the near-end echo-paths and subtracting them from the microphones signals, and communicating the undistorted desired-speech signal to the far-end [2].

The popular normalized least mean-squares (NLMS) adaptive filter is numerically stable and efficient [3,4]. Its sign-error NLMS (SNLMS) variation employs the polarity of the adaptation error [5] and is favorable over the NLMS due to its protection against abrupt noises [6–8]. The adaptation of the SNLMS filter is governed by the step-size parameter, which balances the convergence pace and the adaptation accuracy of the filter. Controlling the step-size is desirable in scenarios of frequent acoustic changes, e.g., echo-path variations and single-to-double-talk transitions. The variable step-size (VSS) problem has motivated Haubner et al. to employ deep learning for near-end speech [9] and noise [10] evaluation, and to reduce the error of the adaptive process [11]. Meta-learning-based solutions have also recently emerged in [12]. The a priori adaptation error and the far-end signal undergo feature extraction for VSS estimate in [13] and a non-parametric VSS (NPVSS) minimized the adaptation error in [14]. The mean-error sigmoid VSS (SVSS) combines adaptation-error history with current adaptation-error estimate [15].

The methods in [9–11] model the far-end signal as linear with its respective echo-signal, and the studies in [13–15] consider the echo path as time-invariant. Unfortunately, both assumptions restrict performance in realistic setups and may cause low adaptation accuracy with slow convergence-pace [16]. On top of that, parameter-tuning, as in the NPVSS [14], involves heuristics that are inaccurate in practice. Thus, SAEC in real-life scenarios remains a relevant challenge and an active research area.

Inspired by [17], we mitigate these disparities by introducing a data-driven framework for deep learning-based VSS (DVSS) that avoids heuristics and does not require acoustic setup hypotheses. First, the update rule of the adaptation process, governed by the step size, integrates the widely-linear model in the complex time domain. The mismatch between the actual echo paths and their filtered estimate is quantified by the normalized misalignment, which is then minimized with respect to the step size. A neural network (NN) relates acoustic signals to the optimal step size in training, and the predicted step size feeds the SNLMS filter in real time for tracking the echo paths. The described framework is novel for SAEC.

We compare our approach with the competition by considering a pair of near-end loudspeakers and microphones, although this framework generalizes to any number of channels. Experimenting with 100 h from the AEC-challenge corpus [18] reveals the consistent advantage of the DVSS in single and double-talk periods across various acoustic setups. The DVSS-SNLMS system also re-converges more rapidly and accurately after abrupt echo-path changes and is more robust to single-to-double-talk transitions.
to the microphones, also known as a room impulse response (RIR). Instead of tracking 4L real-valued coefficients, we turn to the more compact widely-linear model [19] by defining the complex signals:

\[
\begin{align*}
    h(n) &= h_1(n) + j h_2(n), \\
    h^*(n) &= h_1^*(n) + j h_2^*(n),
\end{align*}
\]

where \( j = \sqrt{-1} \), and

\[
\begin{align*}
    h_1(n) &= 0.5 [ h_{LL}(n) + h_{RR}(n) ], \\
    h_2(n) &= 0.5 [ h_{RL}(n) - h_{LR}(n) ], \\
    h_1^*(n) &= 0.5 [ h_{LL}^*(n) - h_{RR}^*(n) ], \\
    h_2^*(n) &= -0.5 [ h_{RL}^*(n) + h_{LR}^*(n) ].
\end{align*}
\]

The complex echo signal \( y(n) = y_l(n) + j y_r(n) \) can now be expressed in a widely-linear manner by \( y(n) = \hat{h}^H(n) \hat{x}_{NL}(n) \):

\[
\begin{align*}
    \hat{h}(n) &= \begin{bmatrix} h(n) \\ h^*(n) \end{bmatrix}, \\
    \hat{x}_{NL}(n) &= \begin{bmatrix} x_{NL}(n) \\ x_{NL}^*(n) \end{bmatrix},
\end{align*}
\]

where \( x_{NL}(n) = x_{NL,L}(n) + j x_{NL,R}(n) \). The superscripts \( H \) and \( * \) correspondingly notate the transpose-conjugate and conjugate operations. As a result, the complex microphone signal \( m(n) = m_L(n) + j m_R(n) \) can be formulated by

\[
m(n) = \hat{h}^H(n) \hat{x}_{NL}(n) + s(n) + w(n),
\]

where \( s(n) = s_L(n) + j s_R(n) \) and \( w(n) = w_L(n) + j w_R(n) \).

The echo estimation \( \hat{y}(n) = \hat{h}(n) \hat{x}(n), \) where \( \hat{x}(n) \) and \( \hat{x}_{NL}(n) \) follow the same notation, is evaluated by tracking the 2L complex-coefficients of \( \hat{h}(n) \) with the SNLMS adaptive filter. Subsequently, the complex near-end speech estimate can be drawn by

\[
\begin{align*}
e(n) &= m(n) - \hat{y}(n) \\
&= (y(n) - \hat{y}(n)) + s(n) + w(n),
\end{align*}
\]

where \( e(n) = e_L(n) + j e_R(n) \). Our focus is two-fold; tracking and cancelling the echo signal, i.e. nullifying \( y(n) - \hat{y}(n) \), and avoiding distortion of the near-end speech, i.e. preserving \( s(n) \).

3. DVSS-SNLMS FILTER FOR SAEC

3.1. Modeling the SNLMS Filter and Step-size in Double-talk

By placing (15) and the definition of \( \hat{y}(n) \) into (16), we respectively derive the a priori and a posteriori errors of the SNLMS filter [4]:

\[
\begin{align*}
e(n) &= \hat{h}^H(n) \hat{x}_{NL}(n) - \hat{h}^H(n-1) \hat{x}(n) + s(n) + w(n), \\
e(n) &= \hat{h}^H(n) \hat{x}_{NL}(n) - \hat{h}^H(n) \hat{x}(n) + s(n) + w(n).
\end{align*}
\]

The update rule of the 2L complex-valued filter coefficients is [5]:

\[
\hat{h}(n) = \hat{h}(n-1) + \mu(n) \hat{x}(n) \text{sign}(e^*(n)),
\]

where \( \hat{h}(0) \) is a column vector of 2L zeros, the step-size is given by \( \mu(n) \in \mathbb{R} \), and \( \text{sign}(z) = z/|z| \) for every \( z \in \mathbb{C} \), where \( | \cdot | \) is the absolute value. From (17)-(19):

\[
e(n) = e(n) - \mu(n) \text{sign}(e(n)) \hat{x}^H(n) \hat{x}(n).
\]

We now force the a posteriori error to a complete echo-cancellation and extract the corresponding expression of the step-size \( \mu(n) \) [19]. Assuming \( s(n) \) and \( w(n) \) are zero-mean and uncorrelated [4]:

\[
\sigma^2(n) = \sigma^2_s(n) + \sigma^2_w(n),
\]

where \( \sigma^2_s(n) = E[|e(n)|^2] \) and \( \sigma^2_w(n) \), \( \sigma^2_s(n) \) follow the same definition. Now, the \( E[| \cdot |^2] \) operator is applied on both sides of (20), and then (21) is substituted into (20). This process yields

\[
\mu(n) = c + \frac{2 \sigma^2_s(n) + \sigma^2_w(n) - \sigma^2_s(n)}{E[|\hat{x}_H(n) \hat{x}(n)|^2]} - c^2,
\]

where \( c = E[|\hat{x}_H(n) \hat{x}(n)|^2]/E[|\hat{x}_H(n) \hat{x}(n)|^2] \).

3.2. Step-size Optimization with a Data-driven Approach

The mismatch between the adaptive and true filter coefficients is often assessed using the normalized misalignment measure [19]:

\[
D(n) = \frac{\| h(n) - \hat{h}(n) \|_2}{\| \hat{h}(n) \|_2},
\]

\[
= \frac{\| h(n) - \hat{h}(n-1) - \mu(n) \hat{x}(n) \text{sign}(e^*(n)) \|_2}{\| \hat{h}(n) \|_2},
\]

Figure 1: Top - SAEC scenario under the widely-linear model. Bottom - the DVSS-SNLMS block, where a NN estimates the step-size \( \mu(n) \) and the SNLMS filter estimates the acoustic paths via \( \hat{h}(n) \).
where (19) was employed to the second transition and $\| \cdot \|_2$ is the $\ell_2$ norm. We now solve a constrained nonlinear optimization problem [20] to yield the optimal step-size. Formally, the normalized misalignment is minimized with respect to the step-size, in dB:

$$\mu^*(n) = \arg \min_{0 < \mu(n) < 2} 20 \log_{10} D(n),$$

(24)

where the condition $0 < \mu(n) < 2$ is dictated by the stability requirements of NLMS-based adaptive filters [4]. The active-set optimization algorithm [21] is utilized to perform optimization. According to (23), the only values involved in deriving $D(n)$ are the far-end and the a priori error signals. This data-driven approach does not require heuristic parameter tuning to estimate $\mu^*(n)$.

3.3. Deep Adaptation to the Optimal Step-size

A deep NN is integrated into our system to model the relation derived in (22) between acoustic signals and the optimal step-size $\mu^*(n)$. Despite the near-end speech and noise signals being inaccessible in reality, the microphone signal can serve as an approximation. Thus, the microphone signal, along with the far-end and a priori error signals, are mapped to their respective step-size. The NN architecture is of a convolutional form [22] with six input channels, one for the real and one for the imaginary part of each of the three input signals. These six waveforms undergo short-time Fourier transform (STFT) [23] separately before being fed to the NN. The specific architecture is standard and follows the one in [17]. During training, optimization is carried to minimize the $\ell_2$ norm between the optimal step-size $\mu^*(n)$ and the output of the network. Inference, the step-size estimate $\mu^*(n)$ is evaluated by the network and injected to an SNLMS filter that tracks the echo paths. Addressing complexity analysis, the NN and the SNLMS filter consume 4.2 Million floating-point operations per second (MFlops) and 4.8 Megabytes (MB) of memory, by employing 1.05 Million parameters. Embedding this system into real-life edge devices for hands-free speech communication is thus considered feasible in terms of resources [24]. One example of dedicated hardware for this task is the NDP120 neural processor by SyntiantSM [25].

4. EXPERIMENTAL SETUP

4.1. Database Acquisition

The database corpus utilized in this study includes 100 h of noisy and clean segments taken from the AEC-challenge [18], where 25 h are simulated recordings, and 75 h are real recordings. The AEC-challenge data involves scenarios with no echo-paths change, where the near-end speaker and devices do not move, and scenarios with echo-paths change, where either the near-end speaker or devices are moving. We consider both double-talk periods and single-talk periods with far-end speakers only. Practically, audio clips are assigned to the original far-end source signal $r(n)$ and to the near-end speech and noise signals, where $s_l(n) = s_r(n)$ and $u_l(n) = u_r(n)$ in this study. To produce the far-end signals $x_l(n)$ and $x_r(n)$, $r(n)$ is randomly propagated via one of 4500 pairs of RIRs that generate $g_l(n)$ and $g_r(n)$, i.e., the acoustic paths between $r(n)$ and the left and right far-end microphones, respectively. To account for realistic acoustic environments, one of 4500 simulated nonlinear functions is applied to every $x_l(n)$ and $x_r(n)$ pair in a random fashion. These nonlinearities are modeled after realistic power amplifiers and loudspeakers in current hands-free hardware [16]. Each pair of nonlinearly-distorted far-end signals $x_{NL,L}(n)$ and $x_{NL,R}(n)$ is randomly propagated via one of 4500 foursomes of near-end RIRs. All RIRs are generated using the Image Method [26] with $L$ coefficients and reverberation times $RT_{60}$, where $RT_{60} \sim U[0.2, 0.5]$ seconds. The near-end stereo-echo-to-speech ratio (SESR) and stereo-echo-to-noise ratio (SENR) levels were drawn from $[-10, 10]$ dB and $[0, 40]$ dB, respectively, where SESR$=10\log_{10}|y(n)|^2 / |s(n)|^2$ and SENR$=10\log_{10}|y(n)|^2 / |w(n)|^2$ in dB [19]. These ratios are derived by running 20 ms frames that overlap by 50%.

4.2. Preprocessing, Training, and Inference

We recognize the well-known non-uniqueness problem in setups of SAEC, where strong coherence between $x_{NL,L}(n)$ and $x_{NL,R}(n)$ may degrade the adaptation process [27]. To mitigate that, we apply the following channel-wise transformation introduced in the context of the widely-linear model [19]. First, we define the positive and negative half-wave rectifiers [28]:

$$x_{NL,L}(n) = x_{NL}(n) + 0.5 [x_{NL}(n) + ||x_{NL}(n)||],$$

(25)

$$x_{NL,R}(n) = x_{NL}(n) + 0.5 [x_{NL}(n) - ||x_{NL}(n)||].$$

(26)

With the element-wise operation $\tan \theta(n) = x_{NL,R}(n) / x_{NL,L}(n)$:

$$x''_{NL,L}(n) = \cos \theta(n) \ |x_{NL}(n)|,$$

(27)

$$x''_{NL,R}(n) = \sin \theta(n) \ |x_{NL}(n)|,$$

(28)

where eqs. (27) and (28) use element-wise arithmetic. This transformation modifies only phase information, so employing $x''_{NL,L}(n)$ and $x''_{NL,R}(n)$ instead of $x_{NL,L}(n)$ and $x_{NL,R}(n)$ allows a desired reduction in coherence with the advantage of little distortion.

The entire 100 h batch of data is split to yield training, validation, and test sets of sizes 80 h, 10 h, and 10 h, respectively. The split is random, but constrained to preserve balance and avoid bias by following the principles in [29]. Using the training and validation parts, the step-size is evaluated once every 800 ms according to (24) with parameter values of $\mu(0) = 3 \times 10^{-5}$ and $L = 2400$. Echo-paths are abruptly changed once every $t$ seconds, where $t \sim U[4, 10]$, which characterizes in-the-wild conversations. Waveforms undergo STFT with running time frames that are 16 ms long and have 50% overlap. Before being inserted into the network, every STFT representation of every channel is attached to its 96 ms past context. Training the network using back-propagation involves learning rate of $10^{-4}$ that decays by $10^{-6}$ every 5 epochs, mini-batch size of 32 ms, and 40 epochs, using Adam optimizer [30]. The real-time inference is done on the test set. After the artificial gain of the network is calibrated according to [31], the step-size estimate is injected from the network output into the SNLMS, which constantly evaluates the echo paths. Training the network took 32 minutes for every 1 h of input data from all channels. The inference time for the end-to-end system, from the network entry to the echo-paths estimate, is 26 ms on average using an Intel Core i7-8700K CPU @ 3.7 GHz with two GPUs of Nvidia GeForce RTX 2080 Ti.
Table 1: Performance with no echo-paths change.

<table>
<thead>
<tr>
<th></th>
<th>SDR [dB]</th>
<th>PESQ</th>
<th>ERLE [dB]</th>
<th>Norm. Mis. [dB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVSS</td>
<td>3.3±0.6</td>
<td>2.45±0.3</td>
<td>16.1±4.8</td>
<td>-17.3±4.4</td>
</tr>
<tr>
<td>NPVSS</td>
<td>2.47±1.0</td>
<td>2.01±0.4</td>
<td>12.9±5.9</td>
<td>-14.8±4.6</td>
</tr>
<tr>
<td>NNVSS</td>
<td>2.41±1.0</td>
<td>1.86±0.5</td>
<td>12.9±6.0</td>
<td>-14.2±4.8</td>
</tr>
<tr>
<td>SVSS</td>
<td>2.12±0.9</td>
<td>1.73±0.5</td>
<td>10.7±6.5</td>
<td>-13.1±4.8</td>
</tr>
<tr>
<td>SNLMS</td>
<td>1.93±1.1</td>
<td>1.60±0.3</td>
<td>9.7±6.8</td>
<td>-11.6±5.2</td>
</tr>
</tbody>
</table>

Table 2: Performance with echo-paths change.

<table>
<thead>
<tr>
<th></th>
<th>SDR [dB]</th>
<th>PESQ</th>
<th>ERLE [dB]</th>
<th>Norm. Mis. [dB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVSS</td>
<td>3.05±0.8</td>
<td>2.27±0.4</td>
<td>10.9±6.3</td>
<td>-12.5±5.7</td>
</tr>
<tr>
<td>NPVSS</td>
<td>2.20±1.2</td>
<td>1.79±0.5</td>
<td>7.9±6.5</td>
<td>-9.9±5.9</td>
</tr>
<tr>
<td>NNVSS</td>
<td>1.98±1.1</td>
<td>1.71±0.5</td>
<td>7.4±6.8</td>
<td>-9.4±6.1</td>
</tr>
<tr>
<td>SVSS</td>
<td>1.91±1.3</td>
<td>1.63±0.5</td>
<td>7.0±6.8</td>
<td>-9.2±5.9</td>
</tr>
<tr>
<td>SNLMS</td>
<td>1.74±1.5</td>
<td>1.51±0.3</td>
<td>6.6±6.3</td>
<td>-8.2±6.0</td>
</tr>
</tbody>
</table>

Table 3: Convergence times [sec] and success rates [%].

<table>
<thead>
<tr>
<th></th>
<th>DVSS</th>
<th>NPVSS</th>
<th>NNVSS</th>
<th>SVSS</th>
<th>SNLMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4s</td>
<td>7.1s</td>
<td>63%</td>
<td>8.5s</td>
<td>55%</td>
<td>8.6s</td>
</tr>
</tbody>
</table>

is defined by $10 \log_{10} \left( \frac{|s(n)|^2}{|e(n) - s(n)|^2} \right)$ and is affected by both echo levels and speech distortion levels. The PESQ we report is the average of the PESQ score between $s_{1}(n)$ and $e_{1}(n)$, and the PESQ score between $s_{2}(n)$ and $e_{2}(n)$. These measures are derived with running time frames of 20 ms with an overlap of 50%. For a complete view of performance, we report the adaptation convergence times and convergence success rates. Convergence is considered achieved when $D(n)$ falls below $-10$ dB and is considered successful if $D(n)$ remains below $-10$ dB until echo-paths change [4].

5. EXPERIMENTAL RESULTS

Our DVSS-SNLMS approach is matched against the VSS approaches in [13–15], correspondingly abbreviated “NNVSS”, “NPVSS”, and “SVSS”. These competing algorithms were integrated with the widely-linear model and the SNLMS filter for an unbiased comparison. The SNLMS filter with step-size of $\mu = 3 \times 10^{-5}$, briefly “SNLMS”, is the classic approach baseline. Performance in Tables 1 and 2 is outlined with mean and standard deviation (std) values, and Table 3 shows average test set values.

We distinguish between the performance when no echo-paths changes occur, i.e., in Table 1, from segments where echo-paths change, as in Table 2. In both cases, we only consider the post-convergence of the adaptive filter. In Table 1 and Table 2, the mean value of the results reflects the advantage of the DVSS method over the competition. The ERLE stresses the leading echo suppression of the DVSS method, and the SDR and PESQ measures reveal its ability to maintain low speech distortion and high speech quality. It is also noted that the low std values of the DVSS indicate the stability of its performance across various acoustic setups. Table 3 affirms that our method achieves the shortest re-convergence times and the most successful convergence rates in scenarios with no echo-path changes. Unlike the competition, our method has shown a prominent ability to track echo paths by adapting the step-size accurately and rapidly while maintaining high robustness and generalization capabilities. This can be associated with our approach avoidance of heuristic parameter tuning and of making acoustic assumptions that often mismatch realistic scenarios.

Fig. 2 depicts the desired convergence behavior of the DVSS-SNLMS filter in a two-fold manner; it shows the most rapid convergence and re-convergence after abrupt echo-paths change, and it is also the least disturbed by the occurrence of double-talk. On the contrary, competing VSS-based methods slightly diverge due to double-talk, which impedes their convergence success afterward.

6. CONCLUSIONS

Controlling the step-size in adaptive filtering can allow for optimally operate between convergence rate and adaptation accuracy. This study attempts to bring this ability a step closer to practice by introducing a general adaptation-control framework that is both non-parametric and does not require acoustic assumptions and apply it to SAEC. Using the widely-linear model, we first derive the optimal step-size by minimizing the filter misalignment in the complex time domain. Then, we train a neural network to predict this optimal step-size from acoustic data in real time. Based on this step-size estimate, the SNLMS filter tracks the echo paths and performs well over competition across various acoustic setups. Future work may focus on generalization to scenarios where near-end microphones capture different versions of the speech and noise signals.
7. REFERENCES


[24] ETSI ES 202 740: *Speech and multimedia Transmission Quality (STQ); Transmission requirements for wideband VoIP loudspeaking and handsfree terminals from a QoS perspective as perceived by the user*, ETSI Std., 2016.


