On the use of the frequency domain prediction error method to improve variable step size and variable regularization algorithms[1]

Jose M. Gil-Cacho*, Toon van Waterschoot‡, Marc Moonen§, and Søren Holdt Jensen¶

*Corresponding author. Katholieke Universiteit Leuven, ESAT-SCD/IBBT-K.U.Leuven Future Health Department, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium, Tel. +32 16321856, Fax +32 16321970, E-mail pepe.gilcacho@esat.kuleuven.be

‡Katholieke Universiteit Leuven, ESAT-SCD/IBBT-K.U.Leuven Future Health Department, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium, Tel. +32 16321927, Fax +32 16321970, E-mail toon.vanwaterschoot@esat.kuleuven.be

§Katholieke Universiteit Leuven, ESAT-SCD/IBBT-K.U.Leuven Future Health Department, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium, Tel. +32 16321060, Fax +32 16 321970, E-mail marc.moonen@esat.kuleuven.be

¶Multimedia Information and Signal Processing Department of Electronic Systems Aalborg University, Fredrik Bajers Vej 7, DK-9220 Aalborg, Denmark, Tel. +45 9940 8654, E-mail shj@es.aau.dk

‖EDICS: AUD-ECHO

Abstract

In this paper we have presented a new framework based on FDAF adaptation and the PEM (FDAF-PEM). We showed that FDAF adaptation helps in reducing the short term cross-correlation between the far-end an near-end signals. We have used the FDAF-PEM framework to improve several VSS and VR algorithms. The FDAF-PEM versions have outperformed the original versions in every simulation. The improvement is achieved both in performance and also in efficiency. In performance, since the best performance among the compared algorithms, both in terms of convergence and DT robustness (overall MSD and curve smoothness) is achieved with PVSS-FDAF-PEM. In terms of efficiency the FDAF-PEM versions are themselves 10 times cheaper compared to the original version.

Index Terms

This research work was carried out at the ESAT Laboratory of Katholieke Universiteit Leuven, in the frame of K.U.Leuven Research Council CoE EF/05/006 'Optimization in Engineering' (OPTEC) and PFV/10/002 (OPTEC), Concerted Research Action GOA-MaNet, the Belgian Programme on Interuniversity Attraction Poles initiated by the Belgian Federal Science Policy Office IUAP P6/04 'Dynamical systems, control and optimization' (DYSCO) 2007-2011, Research Project FWO nr. G.0600.08 'Signal processing and network design for wireless acoustic sensor networks', EC-FP6 project 'Core Signal Processing Training Program' (SIGNAL) and was supported by a Postdoctoral Fellowship of the Research Foundation Flanders (FWO-Vlaanderen, T. van Waterschoot). All the Matlab files (functions and scripts) used to generate the figures can be downloaded from (homes.esat.kuleuven.be/∼pepe). The scientific responsibility is assumed by its authors.

Double talk, Robust Acoustic Echo Cancellation, variable step size, Prediction error method, acoustic echo cancellation, double-talk, frequency domain adaptive filters.

I. INTRODUCTION

Acoustic echo cancellation (AEC) is used in many speech communication systems where the existence of echoes degrades the speech intelligibility and listening comfort [1], [2]. These applications range from mobile or hands-free telephony to teleconferencing or voice over IP (VoIP) services, which are often integrated in smartphones, tablets, notebooks, laptops, etc, which are becoming standard devices in our every-day life. The general set-up for an acoustic echo canceler is depicted in Fig.1. A far-end speech signal $u(i)$ is played back into an enclosure (i.e., the room) through a loudspeaker. In the room there is a microphone to record a near-end speech signal which is to be transmitted back to the far-end side. An acoustic echo path between the loudspeaker and the microphone exists so that the recorded microphone (i.e., desired) signal $d(i) = y(i) + v(i) + n(i)$ contains an undesired echo $y(i)$ plus the near-end speech $v(i)$ and the background noise $n(i)$. The echo signal $y(i)$ can be considered as the far-end signal $u(i)$ filtered by the echo path i.e., the loudspeaker-enclosure-microphone (LEM) response. An acoustic echo canceler seeks to cancel the echo signal component $y(i)$ in the microphone signal $d(i)$, ideally leading to an *echo-free* error (or residual) signal $e(i)$ which is then transmitted to the far-end side. This is done by subtracting an estimate of the echo signal $\hat{y}(i)$ from the microphone signal, i.e., $e(i) = d(i) - \hat{y}(i)$. Standard approaches to AEC rely on the assumption that the echo path can

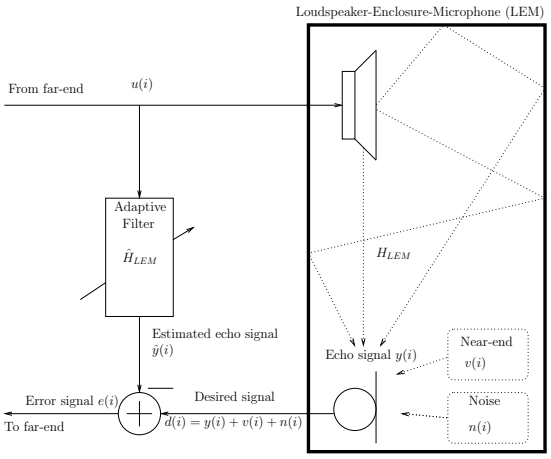


Fig. 1: Acoustic echo canceler set-up

be modeled by a linear FIR filter [1], [2], [3]-[5]. The coefficients of the echo path are collected in the parameter vector $\mathbf{h} = [h_0, h_2, ..., h_{N-1}]^T \in \mathbb{R}^N$. An adaptive filter is used to provide an estimate $\hat{\mathbf{h}} \in \mathbb{R}^N$ of $\mathbf{h}$, such that the echo signal estimate at the microphone position is $\hat{y}(i) = \hat{\mathbf{h}}^T(i)\mathbf{u}(i)$ with $\mathbf{u}(i) = [u(i), u(i-1), \ldots, u(i-N+1)]^T$. Notice that, for now, we also assume that the echo path is time-invariant, $\mathbf{h}^T(i) = \mathbf{h}^T, \forall i$ over the estimation time window $[1, 2, \ldots, t]$ with $t >> N$; we will consider time-varying echo paths further on. A regression data model may be constructed as

$$\mathbf{d}_{\text{LS}}(t) = \mathbf{U}_{\text{LS}}(t)\mathbf{h} + \mathbf{e}_{\text{LS}}(t) \tag{1}$$

where

$$\mathbf{d}_{\text{LS}}(t) = [d(t), d(t-1), ..., d(1))]^T \tag{2}$$

$$\mathbf{e}_{\text{LS}}(t) = [e(t), e(t-1), ..., e(1))]^T \tag{3}$$

$$\mathbf{U}_{\text{LS}}(t) = [\mathbf{u}(t), \mathbf{u}(t-1), ..., \mathbf{u}(1)]^T \tag{4}$$

The error in (1) is to be minimized, which is typically done in a least squares sense,

$$\hat{\mathbf{h}}(t) = \arg\min_{\hat{\mathbf{h}}(t)} \sum_{j=1}^{t} \left| d(j) - \hat{\mathbf{h}}^T(j)\mathbf{u}(j) \right|^2 \tag{5}$$

leading to the well-known least squares (LS) estimate

$$\hat{\mathbf{h}}(t) = \left(\mathbf{U}_{\text{LS}}^T(t)\mathbf{U}_{\text{LS}}(t)\right)^{-1} \mathbf{U}_{\text{LS}}^T(t)\mathbf{d}_{\text{LS}}(t) \tag{6}$$

where $\frac{1}{t}\mathbf{U}_{\text{LS}}^T(t)\mathbf{U}_{\text{LS}}(t) = \mathbf{R}_{uu}(t)$ is the sample autocorrelation matrix and $\frac{1}{t}\mathbf{U}_{\text{LS}}^T(t)\mathbf{d}_{\text{LS}}(t) = \mathbf{r}_{du}(t)$ is the sample cross-correlation vector. Nearly all linear adaptive filtering algorithms are related to the LS estimate which is unbiased but not necessarily minimum variance. Except when the near-end signal is white noise the LS estimate is suboptimal.

In most AEC applications the near-end signal is considered to be uncorrelated with the loudspeaker signal which is an approximation of reality. Moreover, practical AEC implementations rely on computationally simple stochastic gradient algorithms. Therefore, it turns out that the presence of a near-end signal, in a so called double-talk (DT) scenario, will affect the adaptation by making the adaptive filter converge slowly and even diverge. One approach to DT-robust AEC is to equip the (stochastic gradient) adaptive filter with a variable step size (VSS). Some algorithms which utilize some properties of the gradient vector to control the step size have been proposed for DT-robust AEC. From this class of algorithms the projection correlation VSS (PC-VSS) has been proposed in [6]. PC-VSS is chosen as

one of the algorithms in this paper and further explanation will be given. Some example of other VSS algorithms are the ones that stem from the so called non-parametric VSS (NPVSS) proposed in [7]. One such algorithm is the one proposed in [8] so called practical variable step size (PVSS). PVSS is one of the other chosen algorithm and will be further explained.

Another successful method to obtain DT robustness is to remove some correlation in the microphone signal [9] or echo-compensated signal [10] during DT. Also related to this family, in [3] and [11] the aim was to develop a recursive identification algorithm that behaves in a robust way in a continuous DT as well as in a bursting DT scenario based on the prediction of the near-end signal component in the microphone signal. This result is obtained by searching for the optimal AEC solution in a minimum-variance linear estimation framework, rather than in a traditional least squares (LS) framework. The minimum-variance echo path estimate, which is also known as the best linear unbiased estimate (BLUE) [12] depends on the near-end signal characteristics, which are in practice unknown and time-varying [13]. The algorithms in [3] and [11] aim to whiten the near-end signal component in the microphone signal by using adaptive decorrelation filters that are estimated concurrently with the acoustic echo path. The use of the prediction error method (PEM) theory [14] was proposed to estimate the RIR and an autoregressive model of the near end signal. This was previously studied in the context of AFC [13] and a series of algorithms were proposed. Particularly interesting is the prediction-error-method-based adaptive filtering using row operation (PEM-AFROW) [15] which efficiently uses the Levison-Durbin algorithm to estimate both the near-end signal autoregressive coefficients and near-end excitation signal variance.

The minimum variance characteristic is achieved by adding a scaled version of the near-end excitation signal variance to the denominator of the stochastic gradient update equation, i.e., like a variable regularization parameter [16], [17]. For this reason we found interesting to also add a variable regularization algorithm to our list of algorithms and therefore [16] and [17] will be explained further on.

All these algorithms assume that the far-end and the near-end signals are uncorrelated. When a near-end signal is present the LS estimate becomes

$$\hat{\mathbf{h}}(t) = \mathbf{R}_{uu}^{-1}(t)\mathbf{r}_{du}(t) = \mathbf{R}_{uu}^{-1}(t)\mathbf{r}_{yu}(t) + \mathbf{R}_{uu}^{-1}(t)\mathbf{r}_{vu}(t) \tag{7}$$

Usually, the far-end speech and the near-end speech are statistically independent. However In reality the near-end signal is usually speech signal or highly colored noise (e.g., the so called babble noise). Still in the AEC context the second term of (7) is assumed to be zero for the LS estimate to be unbiased. However, we should notice that this assumption is valid only asymptotically and applies to infinitely

long signals. So the question would be what is the actual short-term cross-correlation between these two signals and how much it affect the adaptation process in a stochastic gradient algorithm. We will show in Section III that the instantaneous cross-correlation can be minimized using a transform domain-type of adaptation.

The aim of this paper is therefore two-fold: first, to show why FDAF-PEM is better than PEM itself in dealing with DT situation involving speech signals, and second that by introducing this framework, several VSS and VR can be greatly improved in a cheaper and more efficient manner. As the rest of the paper is full of acronyms we give here a brief guide: the PEM-AFROW acronyms becomes just PEM for simplicity. If the PEM is updated using the APA then it would be written as APA-PEM. The transform domain algorithm using the NLMS or the PEM is written as TD-NLMS or TD-PEM respectively. The same is written if we use a FDAF with a NLMS update or using PEM, i.e., FDAF-NLMS or FDAF-PEM. If other type of VSS is used then the acronyms of these are placed in front, e.g. PVSS-FDAF-PEM.

The paper is organized as follows: Section II explains the prediction error method that is the cornerstone on which the paper is developed. In Section II, we will show the decorrelation properties of FDAF-NLMS, TD-NLMS as compared with NLMS. We will also show the benefits of using FDAF-PEM as a framework where to develop other algorithms. Section IV present the algorithms that are using the proposed framework. Section V present our simulation results and finally Section VI concludes the paper.

II. Prediction error method

We will consider a common approach in PEM-based AFC that is to model the near-end signal with a linear prediction (LP) model linear model][13] for the near-end signal $v(i)$ as

$$v(i) = \frac{1}{A(q,i)} w(i) \tag{8}$$

where q denotes the time-shift operator, e.g., $q^{-k}u(i) = u(i-k)$, and $w(i)$ represents the excitation signal generating the near-end signal. It is assumed to be a white noise signal with time-dependent variance $\sigma_w^2(i)$ i.e.,

$$\mathcal{E}\{w(i)w(i-k)\} = \sigma_w^2(i)\delta(k) \tag{9}$$

where $\mathcal{E}\{\cdot\}$ is the expected value operator. As outlined before, an unbiased identification of the echo path model can be achieved by first prefiltering the loudspeaker signal $u(i)$ and the microphone signal $y(i)$ with the inverse near-end signal model, i.e., $A(q,i)$ before feeding these signals to the adaptive filtering

algorithm. As $A(q,i)$ is obviously unknown, the near-end signal model and the feedback path model have to be jointly identified using PEM i.e.,

$$y(i) = x(i) + v(i) + n(i) \tag{10}$$

$$= H(q,i)u(i) + \frac{1}{A(q,i)}w(i) + n(i) \tag{11}$$

with $H(q,i)$ defined previously and $A(q,i)$ given as

$$A(q,i) = 1 + a_1(i)q^{-1} + ... + a_{n_A}(i)q^{-n_A} \tag{12}$$

where n_A is the LP model order.

The PEM gives an estimate of the models $H(q,i)$ and $A(q,i)$ by minimization of the prediction error criterion

$$\min_{\hat{\boldsymbol{\vartheta}}(i)} \sum_{k=1}^{t} \varepsilon^2 \left[k, \hat{\boldsymbol{\vartheta}}(i)\right] \tag{13}$$

where the prediction error is defined as

$$\varepsilon \left[k, \hat{\boldsymbol{\vartheta}}(i)\right] = \hat{A}(q,i) \left[y(k) - \hat{H}(q,i)u(k)\right] \tag{14}$$

and the parameter vector estimate $\hat{\boldsymbol{\vartheta}}(i) = \left[\hat{\mathbf{h}}^T(i), \hat{\mathbf{a}}^T(i)\right]^T$ contains the parameters of the feedback or echo path model and the near-end signal model estimates as

$$\hat{\mathbf{h}} = [\hat{h}_0(i), \hat{h}_1(i), ..., \hat{h}_{N-1}(i)]^T, \tag{15}$$

$$\hat{\mathbf{a}} = [\hat{a}_0(i), \hat{a}_1(i), ..., \hat{a}_{n_A}(i)]^T, \tag{16}$$

Notice that throughout the paper we assume a sufficient-order condition for the acoustic path model. A final assumption is that the near-end signal $v(i)$ is short-term stationary, which implies that the near-end signal model $\hat{A}(q,i)$ does not need to be re-estimated at each time instant i. That is, instead of identifying the near-end signal AR coefficients recursively, it can also be identified non-recursively on a batch of loudspeaker and microphone data. This is the idea behind the PEM-AFROW [15] which estimates $A(q,i)$ in a frame-based manner, using a frame length that approximates the stationarity interval of the speech. Both $\hat{\mathbf{a}}(i)$ and $\sigma_w^2(i)$ are efficiently calculated using the Levinson-Durbin recursion. PEM-AFROW (PEM from now on) was originally developed in an AFC framework [15] and applied to a continuous DT AEC scenario in [11]. A formal relationship between the BLUE and the PE estimate of the room impulse response $H(q,i)$ can be derived only if the near-end signal characteristics are assumed to be known. In [11], it was shown that minimization of the PE criterion with respect to $\hat{h}(i)$ only, yields the best

(minimum- variance) linear unbiased estimate of $h(i)$ if the near-end signal $v(i)$ can at each time instant $t \in [1, i]$ be modeled as an AR process of order n_A , if the prefilter $\hat{A}(q, i)$ contains at each time instant the true AR coefficients, $a_k(i)$, $k = 1, ..., n_A$, and if the weight $\sigma_w^{-2}(i)$ is at each time instant equal to the inverse variance of the near-end excitation signal $e(i)$, i.e., $\hat{\sigma}_w^{-2}(i) = \sigma_w^{-2}(i)$.

III. PEM VS TD-PEM VS FD-PEM

In this section a justification on the use of FDAF-PEM is given based on several results, namely efficiency, decorrelating properties and performance during double talk. We will start showing that within their adaptation loops NLMS, transform domain NLMS (TD-NLMS) and frequency domain adaptive filter NLMS (FDAF-NLMS) show different decorrelating properties. We will show this using short time averages of the so-called signal $z(i)$ that would measure the correlation within the adaptation loop between the far-end and the near-end signals.

To this end, we consider the NLMS update with $d(i) = y(i) + v(i)$ as

$$e(i) = y(i) - \hat{\mathbf{h}}^T(i-1)\mathbf{u}(i) + v(i) \tag{17}$$

$$= e_0(i) + v(i) \tag{18}$$

So that similarly the weight-update equation becomes

$$\hat{\mathbf{h}}(i) = \hat{\mathbf{h}}(i-1) + \mu \frac{e(i)}{\alpha + \mathbf{u}^T(i)\mathbf{u}(i)}\mathbf{u}(i) \tag{19}$$

$$= \hat{\mathbf{h}}(i-1) + \mu \frac{e_0(i)}{\alpha + \mathbf{u}^T(i)\mathbf{u}(i)}\mathbf{u}(i) + \mu \frac{v(i)}{\alpha + \mathbf{u}^T(i)\mathbf{u}(i)}\mathbf{u}(i) \tag{20}$$

$$= \hat{\mathbf{h}}_0(i) + \tilde{\mathbf{h}}(i). \tag{21}$$

The error signal that is computed at each iteration would therefore become

$$e(i) = y(i) + v(i) - \hat{\mathbf{h}}_0^T(i-1)\mathbf{u}(i) - \tilde{\mathbf{h}}^T(i-1)\mathbf{u}(i) \tag{22}$$

Logically we want the last term in (22), let's call it $z(i)$, i.e., $z(i) = \tilde{\mathbf{h}}^T(i-1)\mathbf{u}(i)$ to be as small as possible which would mean that $\tilde{\mathbf{h}}^T(i-1)$ and $\mathbf{u}(i)$ are uncorrelated to each other at every instant, ideally $z(i) = 0$ which will preserve the near-end signal $v(i)$. This is obviously not the case in practice. To this end we will show that transform domain algorithms can help in decorrelating this two signals and as a result the term $z(i)$ will be smaller than in the NLMS case. We will consider the recursions given in

$N = 80$	NLMS	TD-NLMS	FDAF-NLMS
White noise	2.2 dB	-10.8 dB	-27.2 dB
Colored noise	3.5 dB	-9.6 dB	-27.0 dB
Speech	4.3 dB	-7.9 dB	-23.7 dB

TABLE I: Correlation as given in (23) between far-end and near-end within a adaptation loop.

appendix A where it is shown how the signal $z(i)$ is calculated in the considered algorithms, namely the NLMS, TD-NLMS and the FDAF-NLMS. As a measure of correlation between the far-end and the near-end signals within an adaptation loop we consider

$$X(\text{dB}) = 10\log_{10}\frac{\sum_{i=1}^{L} z(i)^2}{\sum_{i=0}^{L} v(i)^2} \tag{23}$$

which is the normalized variance of the $z(i)$ signal w.r.t. the variance of the near-end signal of length L. Table I shows the values of X in (23) with $N = 80$ taps which is the length of the impulse response of the system under study. These values are calculated using the recursion, given in appendix A.

As expected the degree of decorrelation achieved with NLMS is lower than that of TD-NLMS and FDAF-NLMS in this order. The higher decorrelation is achieved with FDAF-NLMS. It is also shown that the degree of decorrelation is higher when the near-end is white noise then colored noise and finally speech signal. Table I shows 'relatives' values of a non-standard way of measuring correlation and no reference value is known to be high or low, the important matter is the differences amongst them.

In addition we computed correlation X for different filter lengths and the results are shown in Fig. 2. The decorrelation between the far-end and the near-end signal seems to be much higher in the FDAF-NLMS case than in NLMS. Although FDAF-NLMS outperforms TD-NLMS, the latter also obtains high degrees of decorrelation compared to NLMS. This is the reason why in [18] we also obtained good results in DT when using TD-NLMS. It is shown that increasing the filter length also increases the degree of decorrelation in FDAF-NLMS and TD-NLMS but not in NLMS case. The correlation is always smaller when the near-end is white noise, then colored noise and finally speech signal. However, since the far-end signal is itself speech signal, the differences among them seem to be very small i.e., a couple of dB as Table I shows.

Apart of the decorrelating properties of FDAF-NLMS w.r.t. to the far-end and the near-end signals we show in Fig. 3 the performance in terms of MSD of the PEM, the TD-PEM and the FDAF-PEM to

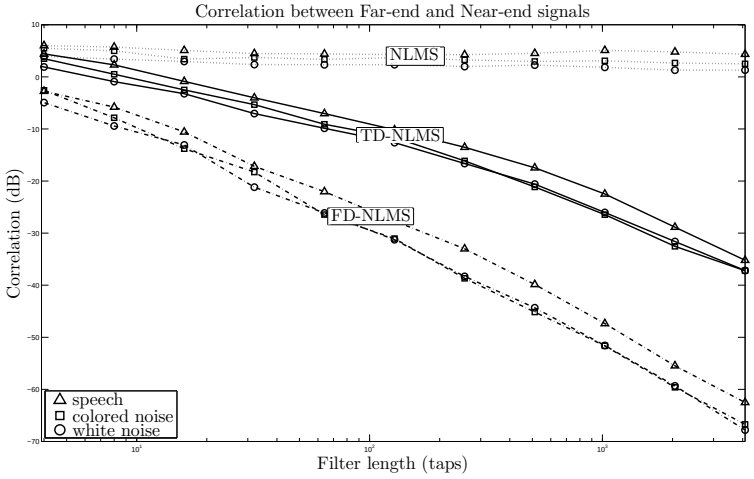
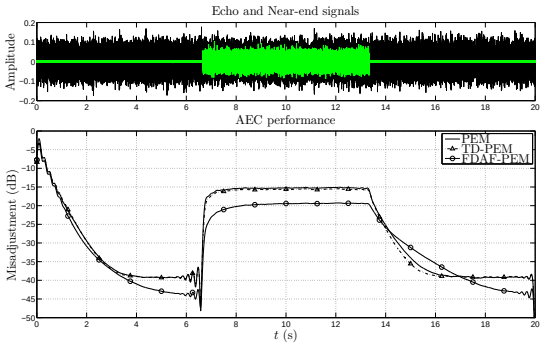
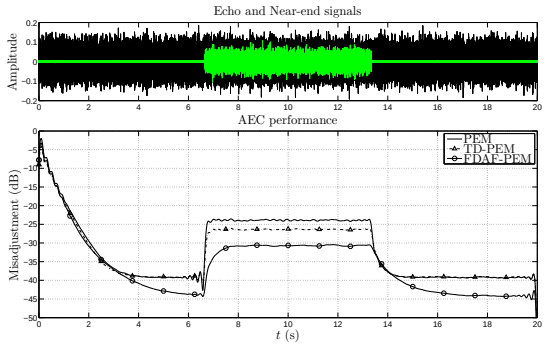


Fig. 2: Correlation between the near-end and the far-end signal within different adaptation loops, namely the NLMS, the transform domain NLMS (TD-NLMS) and the frequency domain adaptive filter using NLMS (FDAF-NLMS). The far-end signal is female speech and the near-end signal is white noise, colored noise or male speech.



(a) White noise as near-end signal. PEM update compared.



(b) Colored noise as near-end signal. PEM update compared.

Fig. 3: AEC performance using colored stationary noise as far-end signal. Several algorithms are compared when different bursting DT signals occur at -10 dB SER and there is white background noise at 30 dB SNR.

compare the performance of actual versions of PEM. We do this with colored noise as far-end and white noise (Fig. 3a) and colored noise (Fig. 3b) as near-end signal and the results of the ensemble average of 100 independent runs. In the white noise case (Fig. 3a) PEM and TD-PEM performs very similar (-15 dB) while FDAF-NLMS obtains -20 dB. In the colored noise case, the TD-NLMS obtains 2 dB more than NLMS although FDAF-NLMS still outperforms both of them by $4-5$ dB in the TD-NLMS case

Algorithm	Computation	Total
PEM	$\left(8+\frac{4P+2n_A+1}{P}\right)N+\frac{1}{P}n_A+\left(4+\frac{4P+2}{P}\right)n_A+\frac{P-1}{P}+10$	980
APA-PEM	$\left(2K^2+4K+2+\frac{4P+2n_A+1}{P}\right)N+4K+\frac{1}{P}n_A+\left(4+\frac{4P+2}{P}\right)n_A+\frac{P-1}{P}+10$	4356
TD-PEM	$\left(8+\frac{4P+2n_A+1}{P}\right)N+\frac{1}{P}n_A+\left(4+\frac{4P+2}{P}\right)n_A+\frac{P-1}{P}$ $+10+6N\log_2 N$	4015
FDAF-PEM	$\frac{18M\log_2 M+18M+n_A^2+n_A(4+4M)}{N}$	307

TABLE II: Complexity comparison by the number of FLOPS per recursion. One FFT/IFFT is a $3M\log_2 M$ complexity operation.

and $5-6$ dB in the NLMS case. Complexity wise FDAF-PEM is also the most efficient as Table II shows: FDAF-PEM is 3 times cheaper than PEM itself and more than 14 times cheaper than TD-NLMS (using FFTs).

Finally Fig. 4 shows the performance of them PEM, the TD-PEM and the FDAF-PEM as well as that of APA-PEM using speech signals as the far-end and the near-end signals. APA-PEM is included for reasons that will become clear latter on. The parameters of the algorithms were tunned to have the same initial convergence rate and acceptable DT robustness. It can be observed that FDAF-PEM obtains the lowest MSD values both in single talk and in DT. Compared to PEM FDAF-PEM achieves $8-9$ dB improvement. APA-PEM seems to improve PEM by a couple of dB during DT. TD-PEM appears to have same MSD during single talk as FDAF-PEM but it is outperformed by FDAF-PEM, during DT by $5-6$ dB. It appears to be clear that for our motivation that is DT robustness we can choose an algorithm that is better and more efficient with FDAF-PEM. Concluding, we depart from PEM and show that FDAF-PEM performs better that the PEM and the TD-PEM. The reason is shown to be the decorrelating property of the FDAF-type of recursion with in the adaption loop. This concludes this section and the discussion on the choice of working framework (i.e. FDAF-PEM) to improve VSS algorithms.

IV. Selection of VSS algorithms

All the three chosen VSS algorithm are defined using APA in their original versions. That is the reason why we also include in this paper the simulation results of PEM when the APA update is used. In this section we show the proposed FDAF-PEM version of the following algorithms: Practical variable step size (PVSS) [19], projection correlation variable step size (PCVSS) [6], variable regularization for affine projection algorithm (VR) [16]. It is important to notice that the FDAF-PEM given in Alg. 1 uses the

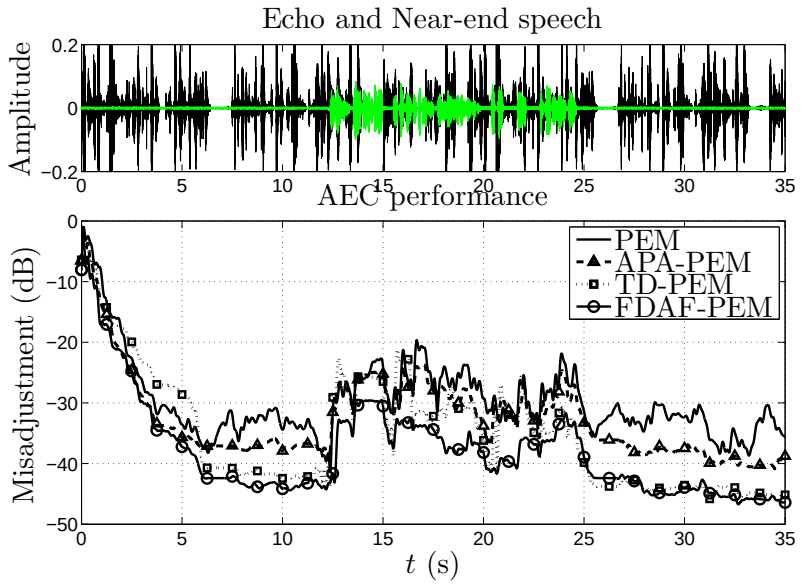


Fig. 4: AEC performance using speech signals (far-end and near-end) between 12.5 and 25 s. PEM, TD-NLMS, FDAF-NLMS and FDAF-PEM are compared when bursting DT occurs at -10 dB SER and white background noise at 30 dB SNR.

inverse estimated variance of the excitation signal $w(i)$, i.e., $\sigma_w^2(i)$ to account for the variance in the estimation as to obtain a BLUE estimator with minimum variance. $\sigma_w^2(i)$ is estimated directly using the Levinson-Durbin algorithm and use it in the variable $\mathbf{G}(\omega, k)$. In the PVSS [19] and VR [17] algorithms the near-end (or excitation in the FDAF-PEM version) signal variance needs to be estimated. FDAF-PEM version employs the same near-end signal variance estimation techniques as those in [19] and [17]. The reason for this is that we wanted to keep the algorithms as comparable as possible.

- Practical VSS [19]

Practical VSS (PVSS), stems from the so called non-parametric VSS proposed in [7]. This algorithm was developed in a system identification context, aiming to recover the system noise from the error of the adaptive filter. Inspired by this idea, several approaches have focused on finding other practical solutions for a real AEC application where the system noise also consists of the near-end signal and consequently, different VSS-NLMS algorithms have been developed e.g., [8] and [20]. Their convergence was slow in practice so, an APA version of [8] was proposed to overcome the limitation of previous algorithms thus deriving in the practical VSS affine projection algorithm (PVSS) [19]. PVSS is claimed to be easy to control in practice and is effectively used in DT situations. The differences resides in the condition that is imposed to preserve the near-end component in the echo-compensated signal. In PVSS the condition

is to set the a posteriori error equal to the near-end signal. As this is not possible in reality, an estimated of the power of the near end signal is calculated and compared to the error signal.

- Projection Correlation VSS [6]

Projection Correlation VSS (PCVSS) belongs to the gradient-based VSS algorithms which have the appealing characteristics of distinguish between echo path change and DT. It is the result of improving the algorithm given in [21] to be specific to DT robust AEC since the latter assumed white input and noise signals. In PC-VSS the adaptation rate is controlled by a measure of the correlation between instantaneous and long-term averages of the, so called projection vectors, i.e., gradient vectors in APA. It appears that PC-VSS outperforms the algorithms given in [21] and [22] in DT situations and moreover it does not rely on any signal or system model so it is easy to control in practice.

- Variable Regularization [16]

In [16] the auhtors proposed a new algorithm to design a variable regularization factor for the APA. They deal with the same type of system identification setup, as in [7]. In order to minimize the difference between the estimated and true adaptive filter coefficients, they assumed that the l_2 norm of the a posteriori error is equal to the noise variance, similar to the condition impossed in [7]. They also compared the performance with the existing techniques [23], [24], [25] and demonstrate the effectiveness of this algorithm.

V. Simulation results

Simulations were performed using speech signals. The sampling frequency in every simulation was 8 kHz. In the AEC simulations the far-end (FE) or loudspeaker signal was a female speech signal and the near-end (NE) signal a male speech signal. The microphone signal consists of three concatenated segments of speech: the first 12.5 s. segment consists of echo only, the second segment is the sum of echo $+$ near-end signal generating a DT situation of 13 s, and the third segment is echo only again. We define two The performance measure *Misadjustment* (MSD) is used to evaluate different VSS algorithms. The MSD between the estimated echo path $\hat{\mathbf{h}}(i)$ and the true echo path $\mathbf{h}$ represents the accuracy of the echo path estimation and is defined as,

$$\text{MSD}(i) = 10\log_{10}\frac{\left\|\hat{\mathbf{h}}(i)-\mathbf{h}\right\|_2^2}{\|\mathbf{h}\|_2^2} \tag{24}$$

These three selected algorithms are also implemented using FDAF-NLMS and FDAF-PEM to show different grades of improvement as compared to the original versions using APA. The reason why

including the FDAF-NLMS version of the algorithms is that in fact, the variable regularization parameter of [17] is exactly the same as that in [16] but it was derived using different frameworks ,i.e., FDAF-NLMS and APA respectively. The results reported in [17] have shown that a simple FDAF implementation of the same variable regularization parameter offered better results. These results and those of [16] were however only reported for stationary (white and colored) noise. We found interesting to show their performance in realistic AEC applications and to compare them with the proposed VR-FDAF-PEM version. Furthermore, prediction error method using APA [26] update (APA-PEM) is also included to show its performance because the other three algorithms use APA updates. The tunning parameters of every algorithm were chosen to have a similar initial convergence that also leads to acceptable DT robustness. The numbers are shown only for the FDAF-PEM versions although the code and therefore the tunning parameters of the other algorithms can be all downloaded from ($homes.esat.kuleuven.be/\sim pepe$). In the algorithms' pseudo code, $\mathcal{F}\{\cdot\}$ and $\mathcal{F}^{-1}\{\cdot\}$ refres to an FFT and IFFT operation respectively, and $(\cdot)^*$ represents a conjugation operation.

Practical VSS (algorithm 2): The initial convergence, shown in Fig. 5(a), of PVSS-FDAF-PEM seems to be as fast as the original version reaching -45 dB of MSD at around 8 s. while surprisingly PVSS-FDAF shows higher MSD -38 dB only. During DT, PVSS-FDAF's curve is surprisingly unaltered but with a higher MSD than that of PVSS-FDAF-PEM. PVSS-FDAF-PEM greatly outperforms PVSS by $8-9$ dB and it shows a very smooth curve with a jump at DT onset and gradually converging back smoothly. Projection Correlation VSS (algorithm 3): The initial convergence, shown in Fig. 5(b), of the three algorithms are similar although PCVSS reaches -45 dB of MSD while PCVSS-FDAF and PCVSS-FDAF-PEM reach 4 dB less. It is during DT where the differences of these algorithms and the improvement of PCVSS-FDAF-PEM w.r.t. the other two algorithms become apparent. PCVSS-FDAF-PEM outperforms the other two algorithms by around $7-8$ dB and it also shows a more stable curve. Variable Regularization (algorithm 4): The initial convergence, shown in Fig. 5(c), of both VR-FDAF and VR-FDAF-PEM is equally fast whereas VR is much slower. The MSD after convergence is different in the three algorithms, the higher MSD is that of VR (-32 dB), then VR-FDAF (-37 dB) and finally VR-FDAF-PEM (-41 dB). During DT VR-FDAF-PEM outperforms the other two setting a lower bound of around $1-2$ dB lower than that of VR and VR-FDAF. As a conclusion, the best performance among the compared algorithms, both in terms of convergence and DT robustness (overall MSD and curve smoothness) is achieved with PVSS-FDAF-PEM. The improvement during DT w.r.t. to the original version seems to be more apparent in the case of PVSS and PCVSS, whereas in the case

of VR the improvement is only marginal. Complexity wise the FDAF versions are all 11 times cheaper that the original versions. The FDAF-PEM versions are themselves 10 times cheaper as they only add 100 FLOPS for a considerable performance increase. Consequently we claim that the PVSS-FDAF-PEM, PCVSS-FDAF-PEM and VR-FDAF-PEM algorithms perform much better than the original versions for DT robustness. Moreover the FDAF version and FDAF-PEM (with a marginal complexity increase) are highly more efficient than the original versions.

VI. Conclusion

In this paper we have presented a new framework based on FDAF adaptation and the PEM (FDAF-PEM). We showed that FDAF adaptation helps in reducing the short term cross-correlation between the far-end an near-end signals. We have used the FDAF-PEM framework to improve several VSS and VR algorithms. The improvement is achieved both in performance and also in efficiency. In performance, since the best performance among the compared algorithms, both in terms of convergence and DT robustness (overall MSD and curve smoothness) is achieved with PVSS-FDAF-PEM. The FDAF-PEM versions have outperformed the original versions in every simulation. In terms of efficiency the FDAF-PEM versions are themselves 10 times cheaper compared to the original version.

Algorithm 1: Frequency Domain implementation of the PEMAFROW [15](FDAF-PEM)

Initialize: $k = 0$, $M = 2N$, $\hat{\mathbf{H}}(\omega, k-1) = \mathbf{e}(k-1) = \boldsymbol{\sigma}_u^2 = \mathbf{0}_{M\times 1}$.

Variables $\mathbf{u}(k)$ and $\mathbf{d}(k)$ are $2N$-length vectors satisfying the overlap-save condition in FDAF, e.g.,

$\mathbf{u}(k) = [u(kN - N + 1), ...u(2kN - N)]$

for $k = 1, 2, ...$ **do**

$\mathbf{e}(k) = \mathbf{d}(k) - \mathcal{F}^{-1}\left\{\mathcal{F}\left\{\mathbf{u}(k)\right\} \cdot \hat{\mathbf{H}}(\omega, k-1)\right\}$

$\mathbf{r}(k) = [\bar{\mathbf{e}}(k) \quad \bar{\mathbf{e}}(k-1)]$ // $\bar{\cdot}$ means *last* N *elements*

$[\mathbf{a}(k) \quad \sigma_w^2(k)] =$ Levison-Durbin $([\mathbf{I}_P \quad \mathbf{0}_{M-P}]\,\mathbf{r}(k), n_A)$

for $m = 1, ..., M$ **do**

$\tilde{u}(m,k) = \mathbf{a}(k)[u(kN+m), ..., u(kN+m-n_A)]^T$

$\tilde{d}(m,k) = \mathbf{a}(k)[d(kN+m), ..., d(kN+m-n_A)]^T$

end for

$\tilde{\mathbf{U}}(\omega, k) = \mathcal{F}\left\{\tilde{\mathbf{u}}(k)\right\}$

$\mathbf{e}(k) = \tilde{\mathbf{d}}(k) - \mathcal{F}^{-1}\left\{\tilde{\mathbf{U}}(\omega, k) \cdot \hat{\mathbf{H}}(\omega, k-1)\right\}$

$\tilde{\mathbf{E}}(\omega, k) = \mathcal{F}\left\{[\mathbf{0}_N \quad \mathbf{I}_N]^T\,\mathbf{e}(k)\right\}$

for $m = 0, 1, ..., M-1$ **do**

$\sigma_u^2(\omega_m, k) = \lambda_0\sigma_u^2(\omega_m, k-1) + (1-\lambda_0)|\tilde{U}(\omega_m, k)|^2$

$G(\omega_m, k) = \left(\sigma_u^2(\omega_m, k) + \sigma_w^2(k) + \alpha\right)^{-1}$

end for

$\hat{\mathbf{H}}(\omega, k) = \hat{\mathbf{H}}(\omega, k-1) + \mu\mathcal{F}\left\{[\mathbf{I}_N \quad \mathbf{0}_N]^T\,\mathcal{F}^{-1}\left\{\mathbf{G}(\omega, k) \cdot \tilde{\mathbf{U}}^*(\omega, k) \cdot \tilde{\mathbf{E}}(\omega, k)\right\}\right\}$

end for

Algorithm 2: PVSS [19] using FDAF-PEM

Initialize: $k = 0$, $M = 2N$, $\hat{\mathbf{H}}(\omega, k-1) = \mathbf{e}(k-1) = \boldsymbol{\sigma}_u^2 = \boldsymbol{\sigma}_e^2 = \boldsymbol{\sigma}_d^2 = \boldsymbol{\sigma}_d^2 = \mathbf{0}_{M\times 1}$.

for $k = 1, 2, ...$ **do**

Perform lines $1-13$ from Algorithm 1

$\tilde{\mathbf{D}}(\omega, k) = \mathcal{F}\left\{[\mathbf{0}_N \quad \mathbf{I}_N]^T\,\tilde{\mathbf{d}}(k)\right\}$

$\tilde{\mathbf{Y}}(\omega, k) = \mathcal{F}\left\{[\mathbf{0}_N \quad \mathbf{I}_N]^T\,\mathcal{F}^{-1}\left\{\tilde{\mathbf{U}}(\omega, k) \cdot \hat{\mathbf{H}}(\omega, k-1)\right\}\right\}$

for $m = 0, 1, ..., M-1$ **do**

$\sigma_e^2(\omega_m, k) = \lambda_0\sigma_e^2(\omega_m, k-1) + (1-\lambda_0)|\tilde{E}(\omega_m, k)|^2$

$\sigma_d^2(\omega_m, k) = \lambda_0\sigma_d^2(\omega_m, k-1) + (1-\lambda_0)|\tilde{D}(\omega_m, k)|^2$

$\sigma_y^2(\omega_m, k) = \lambda_0\sigma_y^2(\omega_m, k-1) + (1-\lambda_0)|\tilde{Y}(\omega_m, k)|^2$

$\sigma_u^2(\omega_m, k) = \lambda_0\sigma_u^2(\omega_m, k-1) + (1-\lambda_0)|\tilde{U}(\omega_m, k)|^2$

$\mu_{\text{PVSS}} = \mu_0\left|1 - \dfrac{\sqrt{|\sigma_d^2(\omega_m, k) - \sigma_y^2(\omega_m, k)|}}{\sqrt{\sigma_e^2(\omega_m, k) + 10^{-8}}}\right|$

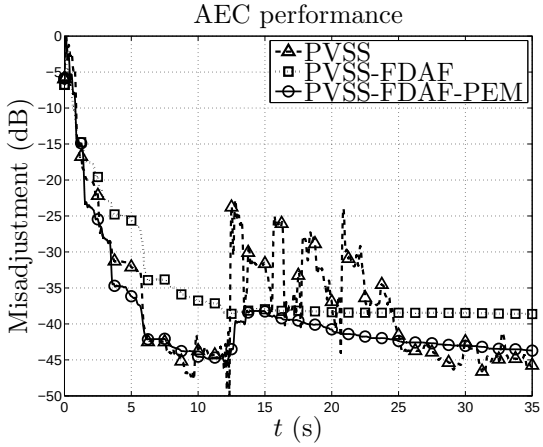
$G(\omega_m, k) = \left(\sigma_u^2(\omega_m, k) + \alpha\right)^{-1}$

end for

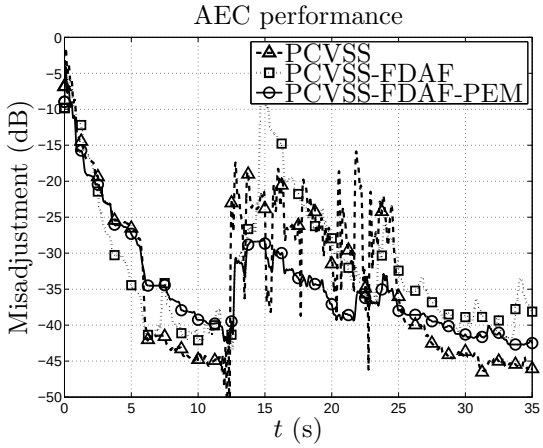
$\hat{\mathbf{H}}(\omega, k) = \hat{\mathbf{H}}(\omega, k-1) + \mathcal{F}\left\{[\mathbf{I}_N \quad \mathbf{0}_N]^T\,\mathcal{F}^{-1}\left\{\boldsymbol{\mu}_{\text{PVSS}}(\omega, k) \cdot \mathbf{G}(\omega, k) \cdot \tilde{\mathbf{U}}^*(\omega, k) \cdot \tilde{\mathbf{E}}(\omega, k)\right\}\right\}$

end for

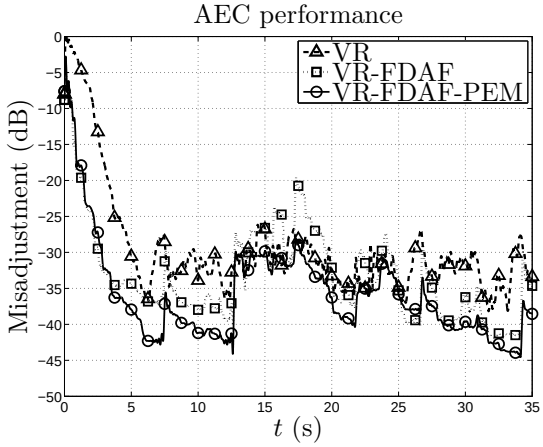
In our simulations we used the following set of parameters: $P = 160$, $N = 80$, $n_A = 1$, $\lambda_0 = 0.999$, $\alpha = 2e^{-6}$, $\mu_0 = 0.0123$.



(a) Practical VSS (PVSS: [19])



(b) Projection Correlation VSS (PCVSS: [6])



(c) Variable Regularization (VR: [16] and VR-FDAF: [17])

Fig. 5: AEC performance using speech signals (far-end and near-end) between 12.5 and 25 s. The original, FDAF and the FDAF-PEM versions are compared when bursting DT occurs at -10 dB SER and white background noise at 30 dB SNR

Algorithm 3: PCVSS [6] using FDAF-PEM

Initialize: $k = 0$, $M = 2N$ and $\hat{\mathbf{H}}(\omega, k-1) = \mathbf{e}(k-1) = \mathbf{0}_{M\times 1}$ and $\mathbf{\Theta}(\omega, k-1) = \mathbf{0}_{M\times B}$.
for $k = 1, 2, ...$ **do**
  Perform lines $1-13$ from Algorithm 1
  $\mathbf{\Phi}(\omega, k) = \tilde{\mathbf{U}}^*(\omega, k) \cdot \tilde{\mathbf{E}}(\omega, k)$
  $\mathbf{\Theta}(\omega, k) = [\mathbf{\Phi}(\omega, k), \mathbf{\Phi}(\omega, k-1), ..., \mathbf{\Phi}(\omega, k-B+1)]$
  for $m = 0, 1, ..., M-1$ **do**
    $\sigma_u^2(\omega_m, k) = \lambda_0 \sigma_u^2(\omega_m, k-1) + (1-\lambda_0)|\tilde{U}(\omega_m, k)|^2$
    $C(\omega_m, k) = \dfrac{1}{B}\sum_{j=1}^{B} \mathbf{\Theta}(\omega_m, j)$
    $\mu'(\omega_m, k) = \lambda_1\ \mu_{\text{PCVSS}}(\omega_m, k-1) + \beta_0\ C(\omega_m, k)$
    $\mu_{\text{PCVSS}}(\omega_m, k) = \begin{cases} 0, & \mu'(\omega_m, k) < 0 \\ \mu_0, & \mu'(\omega_m, k) > \mu_0 \\ \mu'(\omega_m, k), & \text{otherwise} \end{cases}$
    $G(\omega_m, k) = \left(\sigma_u^2(\omega_m, k) + \alpha\right)^{-1}$
  end for
  $\hat{\mathbf{H}}(\omega, k) = \hat{\mathbf{H}}(\omega, k-1) + \mathcal{F}\left\{[\mathbf{I}_N \quad \mathbf{0}_N]^T \mathcal{F}^{-1}\left\{\boldsymbol{\mu}_{\text{PCVSS}}(\omega, k) \cdot \mathbf{G}(\omega, k) \cdot \mathbf{\Theta}(\omega, k)\right\}\right\}$
end for
In our simulation we used the following set of parameters: $P = 160$, $N = 80$, $n_A = 1$, $\lambda_0 = 0.95$, $B = 5$, $\beta = \lambda_1 = 0.9$, $\alpha = 1e^{-3}$, $\mu_0 = 0.03$.

Algorithm 4: Variable Regularization [16], [17] using FDAF-PEM

Initialize: $k = 0$, $M = 2N$, $\hat{\mathbf{H}}(\omega, k-1) = \mathbf{e}(\omega, k-1) = \mathbf{0}_{M\times 1}$ and $\boldsymbol{\sigma}_u^2 = \boldsymbol{\sigma}_e^2 = \boldsymbol{\sigma}_{ue} = \mathbf{0}$.
for $k = 1, 2, ...$ **do**
  Perform lines $1-13$ from Algorithm 1
  $\mathbf{\Phi}(\omega, k) = \tilde{\mathbf{U}}^*(\omega, k) \cdot \tilde{\mathbf{E}}(\omega, k)$
  for $m = 0, 1, ..., M-1$ **do**
    $\sigma_u^2(\omega_m, k) = \lambda_0 \sigma_u^2(\omega_m, k-1) + (1-\lambda_0)|\tilde{U}(\omega_m, k)|^2$
    $\sigma_{ue}(\omega_m, k) = \lambda_1\ \sigma_{ue}(\omega_m, k) + (1-\lambda_1)\,|\Phi(\omega_m, k)|$
    $\sigma_e^2(\omega_m, k) = \lambda_1\ \sigma_e^2(\omega_m, k) + (1-\lambda_1)\left|\tilde{E}(\omega_m, k)\right|^2$
    $\bar{\sigma}_w^2(\omega_m, k) = \left|\sigma_e^2(\omega_m, k) - \dfrac{(\sigma_{ue}(\omega_m, k))^2}{\sigma_u^2(\omega_m, k)}\right|$
    $\alpha(\omega_m, k) = \sigma_u^2(\omega_m, k)\dfrac{\sqrt{\bar{\sigma}_w^2(\omega_m, k)}}{\sqrt{\sigma_e^2(\omega_m, k)} - \sqrt{\bar{\sigma}_w^2(\omega_m, k)}}$
    $G(\omega_m, k) = \left(\sigma_u^2(\omega_m, k) + \alpha(\omega_m, k)\right)^{-1}$
  end for
  $\hat{\mathbf{H}}(\omega, k) = \hat{\mathbf{H}}(\omega, k-1) + \mathcal{F}\left\{[\mathbf{I}_N \quad \mathbf{0}_N]^T \mathcal{F}^{-1}\left\{\mu_0 \cdot \mathbf{G}(\omega, k) \cdot \mathbf{\Phi}(\omega, k)\right\}\right\}$
end for
In our simulation we used the following set of parameters: $P = 160$, $N = 80$, $n_A = 1$, $\mu_0 = 0.065$, $\lambda_0 = 0.95$, $\lambda_1 = 0.9$.

Algorithm	Computation	Total
PVSS	$2K^2N + 4KN + 15K + 8$	3908
PVSS-FDAF	$\frac{21M \log_2 M + 34M}{N}$	375
PVSS-FDAF-PEM	$\frac{6M \log_2 M + 2M + n_A^2 + (4 + 4M)n_A}{N} + 375$	475
PCVSS	$2K^2N + 4KN + 4K + 4N + 12$	4188
PCVSS-FDAF	$\frac{15M \log_2 M + 19M + BM}{N}$	267
PCVSS-FDAF-PEM	$\frac{6M \log_2 M + 2M + n_A^2 + (4 + 4M)n_A}{N} + 267$	367
VR	$2K^2N + 4KN + 4K$	3878
VR-FDAF	$\frac{15M \log_2 M + 31M}{N}$	281
VR-FDAF-PEM	$\frac{6M \log_2 M + 2M + n_A^2 + (4 + 4M)n_A}{N} + 281$	381

TABLE III: Complexity comparison by the number of FLOPS per recursion. One FFT/IFFT is $3M \log_2 M$ FLOPS, one $\sqrt{\cdot}$ is M FLOPS. $K = 4$ which is not in the text.

APPENDIX A
CORRELATION CALCULATION WITHIN THE ADAPTATION LOOP IN THE NLMS, TD-NLMS AND FD-NLMS.

The following recursions are shown to shed some light on the exiting correlation and the effect it has on the adaptation process in different adaptive filtering techniques namely NLMS, TD-NLMS and FDAF-NLMS. It is shown that the signal $z(i)$ in their respective loops is different. The parameters were chosen as $\lambda = 0.99$, $\alpha_0 = N\sigma_u^2$, $\alpha_1 = \frac{N}{2}\sigma_u^2$ with σ_u^2 being the variance of the input signal. The recursion within the NLMS algorithm is given as

$$g(i) = \frac{1}{\mathbf{u}^T(i)\mathbf{u}(i) + \alpha_0} \tag{25}$$

$$\tilde{\mathbf{h}}_{\text{NLMS}}(i) = \lambda\tilde{\mathbf{h}}_{\text{NLMS}}(i-1) + g(i)\mathbf{u}(i)v(i) \tag{26}$$

$$z_{\text{NLMS}}(i) = \tilde{\mathbf{h}}_{\text{NLMS}}^T(i-1)\mathbf{u}(i), \tag{27}$$

similarly the recursion in the TD-NLMS is given as,

$$\mathbf{U}(i) = \mathcal{F}\{\mathbf{u}(i)\} \tag{28}$$

$$P_m(i) = \lambda P_m(i-1) + (1-\lambda)|U_m(i)|^2 \quad m = 1, ..., N \tag{29}$$

$$G_m(i) = (P_m(i) + \alpha_0)^{-1} \quad m = 1, ..., N \tag{30}$$

$$\tilde{\mathbf{H}}_{\text{TD}}(i) = \lambda\tilde{\mathbf{H}}_{\text{TD}}(i-1) + \mathbf{G}(i)\cdot\mathbf{U}(i)\cdot v(i) \tag{31}$$

$$z_{\text{TD}}(i) = \tilde{\mathbf{H}}_{\text{TD}}^H(i-1)\mathbf{U}(i), \tag{32}$$

where $(\cdot)^H$ represents a conjugate transpose respectively. Finally the recursion in the FDAF-NLMS is written as,

$$\mathbf{U}(k) = \mathcal{F}\left\{[\mathbf{u}(k-1) \quad \mathbf{u}(k)]\right\} \tag{33}$$

$$\mathbf{V}(k) = \mathcal{F}\left\{[\mathbf{0}_N \quad \mathbf{v}(k)]\right\} \tag{34}$$

$$P_m(k) = \lambda P_m(k-1) + (1-\lambda)|U_m(i)|^2 \quad m = 1, ..., 2N \tag{35}$$

$$G_m(k) = (P_m(k) + \alpha_0)^{-1} \quad m = 1, ..., 2N \tag{36}$$

$$\tilde{\mathbf{H}}_{\text{FD}}(k) = \lambda\tilde{\mathbf{H}}_{\text{FD}}(k-1) + \mathcal{F}\left\{[\mathbf{I}_N \quad \mathbf{0}_N]^T\mathcal{F}^{-1}\left\{\mathbf{G}(k)\cdot\mathbf{V}(k)\cdot\mathbf{U}^*(k)\right\}\right\} \tag{37}$$

$$\mathbf{z}_{\text{FD}}(k) = [\mathbf{I}_N \quad \mathbf{0}_N]^T\mathcal{F}^{-1}\left\{\tilde{\mathbf{H}}_{\text{FD}}(k-1)\cdot\mathbf{U}(k)\right\}. \tag{38}$$

where $(\cdot)^*$ represents a conjugate operation.

REFERENCES

[1] J. Benesty, T. Gänsler, D. R. Morgan, M. M. Sondhi, and S. L. Gay, *Advances in Network and Acoustic Echo Cancellation*. Berlin: Springer-Verlag, 2001.

[2] P. Gil-Cacho, T. van Waterschoot, M. Moonen, and S. H. Jensen, "Multi-microphone acoustic echo cancellation using multi-channel warped linear prediction of common acoustical poles," in *Proc. 18th European Signal Process. Conf. (EUSIPCO10)*, Aalborg, Denmark, Aug. 2010, pp. 2121–2125.

[3] T. van Waterschoot, G. Rombouts, P. Verhoeve, and M. Moonen, "Double-talk-robust prediction error identification algorithms for acoustic echo cancellation," *IEEE Trans. Signal Process.*, vol. 55, no. 3, pp. 846–858, Mar. 2007.

[4] S. Haykin, *Adaptive Filter Theory*. Upper Saddle River, NY: Prentice Hall, 2002.

[5] P. S. R. Diniz, *Adaptive filtering: Algorithms and Practical Implementations*. Boston, MA: Springer, 2008.

[6] T. Creasy and T. Aboulnasr, "A projection-correlation algorithm for acoustic echo cancellation in the presence of double talk," in *Proc. 2000 IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP'00)*, vol. 1, Istanbul, Turkey, June 2008, pp. 436–439.

[7] J. Benesty, H. Rey, L. R. Vega, and S. Tressens, "A nonparametric VSS NLMS algorithm," *IEEE Signal Process. Lett.*, vol. 13, no. 10, pp. 581–584, Oct. 2006.

[8] C. Paleologu, S. Ciochină, and J. Benesty, "Double-talk robust vss-nlms algorithm for under-modeling acoustic echo cancellation," in *Proc. 2008 IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP'08)*, Las Vegas, USA, Mar. 2008.

[9] X. Lu and B. Champagne, "Pitch analysis-based acoustic echo cancellation over a nonlinear channel," in *Proc. 11th European Signal Process. Conf. (EUSIPCO'02)*, Toulouse, France, Sep. 2002.

[10] X. Hu, A.-Q. Hu, Y. Chen, and X.-H. Zeng, "An adaptive acoustic echo cancellation without double talk detection," in *Proc. 2003 Int. Conf. Machine Learning Cybern. (ICMLC'03)*, Xian, China, Nov. 2003.

[11] T. van Waterschoot and M. Moonen, "Double-talk robust acoustic echo cancellation with continuous near-end activity," in *Proc. 13th European Signal Process. Conf. (EUSIPCO '05)*, Antalya, Turkey, Sep. 2005.

[12] S. M. Kay, *Fundamentals of statistical signal processing: Estimation theory*. Upper Saddle River, New Jersey: Prentice Hall, 1993.

[13] A. Spriet, I. Proudler, M. Moonen, and J. Wouters, "Adaptive feedback cancellation in hearing aids with linear prediction of the desired signal," *IEEE Trans. Signal Process.*, vol. 53, no. 10, pp. 3749–3763, Oct. 2005.

[14] L. Ljung, *System Identification: Theory for the user*. Englewood Cliffs, New Jersey: Prentice Hall, 1987.

[15] G. Rombouts, T. van Waterschoot, K. Struyve, and M. Moonen, "Acoustic feedback cancellation for long acoustic paths using a nonstationary source model," *IEEE Trans. Signal Process.*, vol. 54, no. 9, pp. 3426–3434, Sep. 2006.

[16] W. Yin and A. S. Mehr, "A variable regularization method for affine projection algorithm," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 6, pp. 476 – 480, June 2010.

[17] A. Faza, S. Grant, and J. Benesty, "Adaptive regularization in frequency-domain NLMS filters," in *Proc. 20th European Signal Process. Int. Conf. (EUSIPCO'12)*, Bucharest, Rumania, Aug. 2012, pp. 2625–2628.

[18] J. M. Gil-Cacho, T. van Waterschoot, M. Moonen, and S. H. Jensen, "Transform domain prediction error method for improved acoustic echo and feedback cancellation," in *Proc. 20th European Signal Process. Int. Conf. (EUSIPCO'12)*, Bucharest, Rumania, Aug. 2012, pp. 2422–2426.

[19] C. Paleologu, J. Benesty, and S. Ciochină, "A variable step-size affine projection algorithm designed for acoustic echo cancellation," *IEEE Trans. Audio Speech Lang. Process.*, vol. 16, no. 8, pp. 1466 – 1478, Nov. 2008.

[20] M. A. Iqbal and S. L. Grant, “Novel variable step size NLMS algorithms for echo cancellation,” in *Proc. 2008 IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP’08)*, Las Vegas, USA, Mar. 2008.

[21] V. J. Mathews and Z. Xie, “A stochastic gradient adaptive filter with gradient adaptive step size,” *IEEE Trans. Signal Process.*, vol. 41, no. 6, pp. 2075–2087, June 1993.

[22] C. Rohrs and R. Younce, “Double talk detector for echo canceler and method,” US Patent US 4 918 727, 04 17, 1990. [Online]. Available: http://www.patentlens.net/patentlens/patent/US4918727/

[23] V. Myllylä and G. Schmidt, “Pseudo-optimal regularization for affine projection algorithms,” in *Proc. 2002 IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP’02)*, vol. 2, Orlando, Florida, May 2002, pp. 1917–1920.

[24] H. Rey, L. R. Vega, S. Tressens, and J. Benesty, “Variable explicit regularization in affine projection algorithm: Robustness issues and optimal choice,” *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 2096–2108, May 2007.

[25] Y.-S. Choi, H.-C. Shin, and W.-J. Song, “Adaptive regularization matrix for affine projection algorithm,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 54, no. 12, pp. 1087–10 916, Dec. 2007.

[26] K. Ozeki and T. Umeda, “An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties,” *Electronics and Communication in Japan*, vol. 67, no. 5, pp. 19–27, Aug. 1984.