



Copyright © IEEE.
Citation for the published paper:

Title:

Author:

Journal:

Year:

Vol:

Issue:

Pagination:

URL/DOI to the paper:

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of BTH's products or services Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by sending a blank email message to pubs-permissions@ieee.org.

By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

Low-complexity network echo cancellation approach for systems equipped with external memory

Magnus Berggren, *Student Member, IEEE*, Markus Borgh, *Student Member, IEEE*, Christian Schüldt, *Student Member, IEEE*, Fredric Lindström, Ingvar Claesson, *Member, IEEE*

Abstract— Long delays and sparseness characterize impulse responses in telecommunication networks and a vast number of solutions for network echo cancellation have been proposed over the years. In this paper, an approach for detecting dispersive regions of a sparse impulse response and a proportionate normalized least mean square (PNLMS)-based selective updating approach are combined with an adaptive double-talk detector to form a complete solution for echo cancellation. The proposed solution has low computational complexity and is targeted for systems equipped with external memory.

Index Terms—Network echo cancellation, line echo cancellation, sparse impulse response, NLMS, proportionate NLMS, PNLMS, double-talk detection, DTD.

I. INTRODUCTION

SYSTEMS with sparse impulse response (IR), i.e. containing one or more short dispersive regions separated by regions of flat delays, are common in communication network structures. Figure 1 shows a scenario where several dispersive regions are present. It should be noted that delays as well as the number of dispersive regions can differ significantly from one call to another. In public switched telephone networks (PSTN), typical delays for transcontinental calls are in the order of 30 ms [1] while in IP-networks one-way delays are in the order of 40 to 70 ms [2]. In the extreme case of satellite communication the delay can reach over 1000 milliseconds [3].

Several methods for network echo cancellation which take advantage of the sparse nature of the IR have previously been proposed. Typically, short adaptive filters of proper lengths are placed in time so that they correspond to the dispersive regions. One group of such methods identifies the locations of the dispersive regions by adaptively changing the time location of the filters [4]–[6]. Another set of methods use an auxiliary filter for identification of the dispersive regions [7]–[9]. Since the purpose of the auxiliary filter is only to identify the dispersive regions, complexity reduction methods such as

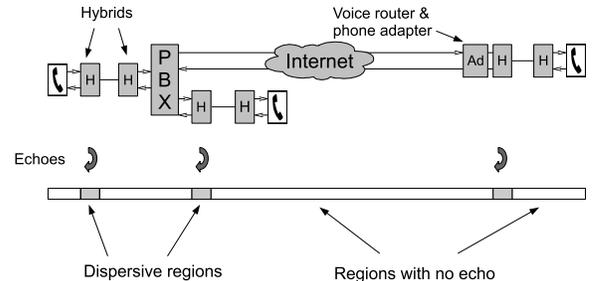


Fig. 1. Call scenario showing a sparse impulse response.

subsampling can be used in the processing of the auxiliary filter [7].

Several algorithms for filter adaptation, making use of the sparseness property, have been proposed, e.g. proportionate normalized least mean square (PNLMS) [10], PNLMS++ [11], improved PNLMS (IPNLMS) [12], improved IPNLMS (IIPNLMS) [13] and an approach by Gänsler, Benesty, Sondhi and Gay (GBSG) [14]. Further, there exist a number of algorithms which reduce the complexity of the adaptive filtering that can be used in combinations with algorithms targeted for sparse IR, e.g. partial update sparse NLMS (PSNLMS) [15] or selective partial update NLMS (SPUNLMS) [16]. Combinations of proportionate filter adaptation algorithms and approaches for reduced complexity, such as selective updating, where the adaptive filter is divided into multiple sections which are adapted at a reduced rate, have also been proposed [17].

A situation where both parties in a telephone call are active simultaneously is called double-talk. Updating adaptive filters during double-talk can lead to filter divergence and poor echo cancellation performance due to interference of the far-end speech signal. Thus, in an echo cancellation application a double-talk detector (DTD) [18]–[20] is normally used to determine when to freeze the update of the adaptive filter(s).

Since the echo canceller is typically not able to completely remove the echo [20], a mechanism for removing the residual echo is required. A common approach is to use a non-linear processor (NLP) [20] controlled by the DTD. The purpose of the NLP is to completely attenuate the echo when only echo is present and restrict the attenuation during double-talk, as the far-end speech will mask parts of the residual echo.

Traditionally, far-end network echoes are handled by network echo cancellers operating in the network so that con-

Copyright (c) 2010 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

M. Berggren, C. Schüldt and I. Claesson are with Blekinge Institute of Technology, Department of Electrical Engineering, SE-37179, Karlskrona, Sweden. (e-mail: magnus.berggren@bth.se; christian.schuldt@bth.se; ingvar.claesson@bth.se)

M. Borgh and F. Lindström are with Limes Audio AB, Tvistevägen 47, SE-90729, Umeå, Sweden. (e-mail: markus.borgh@limesaudio.com; fredric.lindstrom@limesaudio.com)

The funding from the Swedish Knowledge Foundation (KKS) is gratefully acknowledged.

sumer electronic products, such as speakerphones, only need to deal with the direct line echo. However, in some cases, e.g. when an inferior quality telephone is used at the far-end side in combination with a poor network echo canceller, there might still be substantial echoes from the network at the near-end side. (It would of course be desirable to attend to the problems directly at the origin, but this is not always possible in practice from the near-end side perspective.) The uncanceled echoes could be mistaken for far-end speech by the DTD, causing anomalous behavior of the NLP, resulting in poor audio quality and annoyance to the users. Further, the echoes could be significantly delayed due to e.g. packet transmission over the internet. These far-end echoes will typically fall outside the region covered by the line echo canceller of the speakerphone and thus remain uncanceled. The long delays present a challenge for manufacturers of consumer electronic products who aim at providing a solution capable of handling situations where significant echoes are transmitted back from the network.

In consumer electronic products, cost is naturally an issue and thus it is desired to reduce the memory requirements of the used algorithms. Expensive signal processing circuits with internal on-chip memory able to store filters of hundreds of milliseconds are in many cases not an option. However, slower and cheaper external memory might already be available for other purposes, e.g. storing of data (such as sounds and images). Thus, using external memory for storing the filters could be an attractive option.

This paper presents a complete solution for network echo cancellation, addressing the problem of network echoes with long delays, targeted for systems, e.g. consumer electronic devices, with external memory. The solution combines methods for auxiliary filtering and sparse adaptive filtering with a double-talk detector which in turn controls the processing of the residual echo.

The outline of the paper is as follows. In section II the scheme of the proposed system is presented; describing how the long vectors are allocated in external memory. In section III the proposed block-based adaptation of the auxiliary filter is explained in more detail and simulation results verifying the desired properties are shown. Then, in section IV it is shown how the auxiliary filter is used to identify dispersive regions and, finally, section V presents a proposal for how the knowledge of the dispersive regions can be used to control the DTD in order to avoid anomalous behavior while still performing well.

II. PROPOSED SCHEME

The scheme of the proposed method is shown in figure 2. The output signal to the network is denoted $x(n)$ and the input signal from the network is denoted $y(n)$, where n is the sample index. In conventional adaptive filtering a filter $\hat{\mathbf{h}}(n) = [\hat{h}_0(n), \dots, \hat{h}_{L-1}(n)]$, where L is the filter length, is used to model the impulse response [21] and to obtain an echo-cancelled signal. In the proposed system however, a set of short filters are used, (compare with [7]–[9], [22]), to obtain

the echo-cancelled output signal $e(n)$ as

$$e(n) = y(n) - \sum_{i=1}^M \mathbf{x}_i(n)^T \hat{\mathbf{h}}_i(n-1), \quad (1)$$

where M is the number of short filters,

$$\hat{\mathbf{h}}_i(n) = [\hat{h}_{\Delta_i}(n), \dots, \hat{h}_{\Delta_i+L_i-1}(n)]^T \quad (2)$$

denotes the short filter with index i , L_i is the length of the filter and Δ_i is the delay to the start position of the filter and

$$\mathbf{x}_i(n) = [x(n - \Delta_i), \dots, x(n - \Delta_i - L_i + 1)]^T \quad (3)$$

is the corresponding regressor vector. Thus, the filter lengths L_i and delays Δ_i , correspond to the length and position of the dispersive regions, respectively, shown in figure 3, and are determined by a control function using information from the auxiliary filter. (The process of obtaining this information from the auxiliary filter is described in section IV.) The delays Δ_i are implemented using a delay line buffer

$$\mathbf{x}(n) = [x(n), \dots, x(n - L + 1)]^T. \quad (4)$$

In the auxiliary filter processing, the output and input signals $x(n)$ and $y(n)$ are low-pass filtered and decimated by a factor D , yielding the subband signals $x_s(l)$ and $y_s(l)$, respectively, where l is the subband sample index. The subsampled output signal $x_s(l)$ is filtered with the auxiliary filter

$$\hat{\mathbf{h}}_s(l) = [\hat{h}_{s,0}(l), \dots, \hat{h}_{s,L_s-1}(l)]^T, \quad (5)$$

producing an error signal $e_s(l)$. L_s is the auxiliary filter length which should be chosen so that the whole region where an echo may occur is covered. The error signal is used for adaptive update of the auxiliary filter (which is explained in more detail in section III).

In order to allow a reduction of computational complexity by partial filtering and updating, the auxiliary filter is divided into blocks according to

$$\hat{\mathbf{h}}_s(l) = [\hat{\mathbf{h}}_{s,0}(l), \hat{\mathbf{h}}_{s,1}(l), \dots, \hat{\mathbf{h}}_{s,N_B-1}(l)]^T, \quad (6)$$

where N_B is the number of blocks and

$$\hat{\mathbf{h}}_{s,j}(l) = [\hat{h}_{s,jB_s}(l), \dots, \hat{h}_{s,(j+1)B_s-1}(l)]^T, \quad (7)$$

where j is the block index and B_s is the size of the blocks, see figure 3. The subband signal regressor vector $\mathbf{x}_s(l)$ is also divided into blocks of the same length in the same manner. Analogously, the short dispersive region filters $\hat{\mathbf{h}}_i(n)$, the corresponding regressor vectors $\mathbf{x}_i(n)$, as well as the delay line buffer $\mathbf{x}(n)$, are all divided into blocks where the block size is

$$B = DB_s, \quad (8)$$

i.e. the subsampled blocks and the fullband blocks correspond to the same time window. This block structure allows identification of dispersive regions in the auxiliary filter, which can be directly used to place the short adaptive filters in time to match the corresponding dispersive time region.

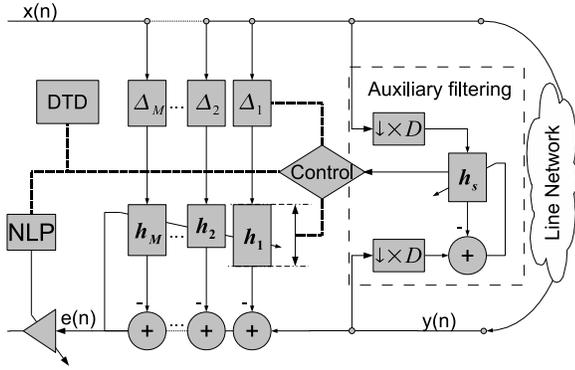


Fig. 2. Schematic diagram of the proposed method.

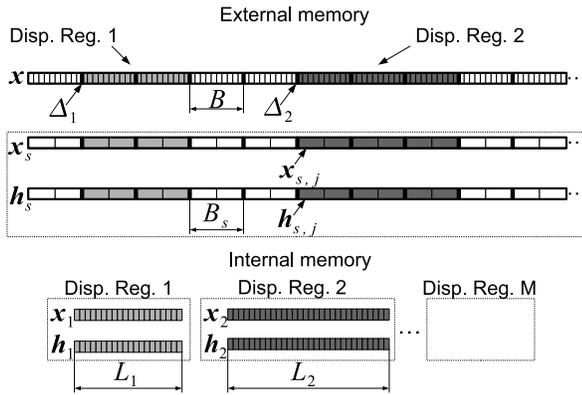


Fig. 3. Memory allocation for delay buffers and filter coefficients.

A. The use of external memory

A long echo delay requires long filter buffers, as described in section I. In order to save expensive internal memory, it is proposed that the long auxiliary filter and the corresponding regressor vector are placed in the external memory, see figure 3. Placement in the external memory will increase the processing complexity, since in many digital signal processing units access to the external memory is time consuming. However, the auxiliary filter is only used to determine the location of the dispersive regions and the processing is thus not time critical. Further, since the IR is mostly stable during a call, periodic update can be used [14], allowing a significant reduction of the computational complexity.

The purpose of the delay line buffer is to host intermediate samples and it is also placed in the external memory. The short adaptive filters $\hat{h}_i(n)$ and their corresponding regressor vectors $\mathbf{x}_i(n)$ are used to produce the output error, i.e. they are used frequently and are thus placed in the internal on-chip memory.

III. ADAPTATION OF THE AUXILIARY FILTER

In this section, proportionate adaptive filtering is summarized, followed by the proposed auxiliary filtering method, which basically is composed of block-partial filtering modifications of existing methods in the PNLMS/IPNLMS family.

A. Proportionate adaptive filtering

PNLMS, PNLMS++(AU) [11] and IPNLMS [12] use proportionate weighting of the filter coefficients to increase the convergence speed of the adaptive filter. The filter update can be generalized as [23]

$$\hat{\mathbf{h}}_s(l) = \hat{\mathbf{h}}_s(l-1) + \frac{\mu \mathbf{G}(l-1) \mathbf{x}_s(l) e_s(l)}{\mathbf{x}_s^T(l) \mathbf{G}(l-1) \mathbf{x}_s(l) + \delta}, \quad (9)$$

where μ is the step size control factor, δ is a regularization parameter and

$$\mathbf{G}(l) = \begin{bmatrix} g_0 & 0 & \dots & 0 \\ 0 & g_1 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & g_{L_s-1} \end{bmatrix}, \quad (10)$$

is the weight matrix that controls the stepsize of the individual filter coefficients.

As a way to reduce the computational complexity of the adaptive filtering, partial updating schemes, such as SC-IPNLMS [17] and BSC-IPNLMS [24], can be used. The proportionate partial update can be generalized as

$$\hat{\mathbf{h}}_s(l) = \hat{\mathbf{h}}_s(l-1) + \frac{\mu \mathbf{U}(l-1) \mathbf{x}_s(l) e_s(l)}{\mathbf{x}_s^T(l) \mathbf{G}(l-1) \mathbf{x}_s(l) + \delta}, \quad (11)$$

where $\mathbf{U}(l) = \mathbf{A}(l) \mathbf{G}(l)$ and where $\mathbf{A}(l)$ is the activity matrix

$$\mathbf{A}(l) = \begin{bmatrix} a_0 & 0 & \dots & 0 \\ 0 & a_1 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & a_{L_s-1} \end{bmatrix}, \quad (12)$$

which defines the active coefficient according to

$$a_i(l) = \begin{cases} 1, & i \in S_a \\ 0, & i \notin S_a \end{cases}, \quad (13)$$

where S_a is the active set which is determined by the used partial update method.

B. Proposed algorithm for adapting the auxiliary filter

Previously proportionate partial update schemes are either not directly designed for line echo cancellation with significant delays [17], [24] and/or not block-based [14], [24], and can therefore not fully utilize the reduction in complexity provided by block-based partial filtering.

Thus, a proportional partial update scheme specifically targeted for sparse systems, where the objective is simply to identify the dispersive regions, is proposed. In this case, it is for instance more important to achieve rapid convergence initially and after echo-path changes than, e.g., to constantly cancel echoes as much as possible, since once the dispersive regions are identified the task of the auxiliary filter is fulfilled.

The proposed partial block updating scheme uses an NLMS update every K :th iteration, and a partial block update otherwise. The i :th $\mathbf{U}(l)$ matrix element for the proposed algorithm is

$$u_{\text{Prop},i}(l) = \begin{cases} g_{\text{Prop},i}(l) a_i(l), & l \neq mK \\ 1, & l = mK \end{cases}, \quad (14)$$

where $m \in \mathbb{Z}$ and the active coefficients weight are set individually according to the IPNLMS scheme as

$$g_{\text{Prop},i} = \frac{1 - \alpha}{2L_s} + (1 + \alpha) \frac{|\hat{h}_{s,i}(l)|}{\|\hat{\mathbf{h}}_s(l)\|_1}, \quad (15)$$

where α is a parameter controlling the relative weighting [12].

The partial update is controlled by the activity set $S_{a,\text{Prop}}$ which is defined by

$$S_{a,\text{Prop}} = \{j_1, \dots, j_{P_B} : \min_{j \in S_{a,\text{Prop}}} \|\hat{\mathbf{h}}_{s,j}\|_\infty > \max_{j \notin S_{a,\text{Prop}}} \|\hat{\mathbf{h}}_{s,j}\|_\infty\}, \quad (16)$$

where P_B is the number of active blocks, resulting in $P_s = B_s P_B$ coefficients to be updated.

To reduce the computational complexity, $\mathbf{x}_s^T(l)\mathbf{G}(l-1)\mathbf{x}_s(l)$ in the denominator of the filter update equation (11) is approximated with $\mathbf{x}_s^T(l)\mathbf{U}(l-1)\mathbf{x}_s(l) + \frac{L_s - P_s}{L_s} \mathbf{x}_s^T(l) \frac{1-\alpha}{2L_s} \mathbf{x}_s(l)$, where the second term accounts for the part of the filter which is not updated. It is thus assumed that all inactive coefficients have the weight $\frac{1-\alpha}{2L_s}$.

To further reduce complexity a partial filtering scheme is proposed where the error signal is calculated as

$$e_{s,\text{Prop}}(l) = \begin{cases} y(l) - \sum_{i=0}^{L_s-1} a_i(l)x_s(l-i)\hat{h}_{s,i}(l-1), & l \neq mK \\ y(l) - \sum_{i=0}^{L_s-1} x_s(l-i)\hat{h}_{s,i}(l-1). & l = mK \end{cases} \quad (17)$$

It is well known that the low complexity, block-based LMS inherently suffers from reduced convergence rate. Also, in a setup where the system can handle echoes which are delayed hundreds of milliseconds, a large filter length is required and long filters imply slower convergence [21]. In order to achieve a solution which can handle the extreme echoes, yet not reduce initial convergence speed in the case where the dispersive regions are located in the first part of the filter, only the first part of the filter is updated during initial convergence, i.e.

$$L_{s,\text{Prop}}(l) = \begin{cases} P_s, & l \leq T_{\text{init}} \\ L_s, & l > T_{\text{init}} \end{cases}, \quad (18)$$

where T_{init} is the number of iterations the initial length is used.

After an echo-path change where a dispersive region becomes a region of flat delay, the filter taps corresponding to that region will adapt towards zero. When the taps reach sufficiently low values they will no longer define an active block. Thereafter, the proportionate update will no longer take place for these taps, leading to a slower convergence. A modification to avoid this sets the taps in an inactive block, which has recently been active, to zero, i.e.

$$\hat{\mathbf{h}}_{s,j}(l) = \mathbf{0}, \quad \text{if } \mathbf{a}_j(l - T_2) = \mathbf{1} \wedge \mathbf{a}_j(k) = \mathbf{0}, \quad (19) \\ \forall k = \{l - T_2 + 1, \dots, l\}$$

where T_2 is the number of iterations the filter block is required to be inactive before a reset is allowed.

TABLE I
COMPUTATIONAL COMPLEXITY MEASURED IN NUMBER OF MULTIPLICATIONS PER ITERATION.

Algorithm	Multiplications
NLMS	$2L_s$
PNLMS	$5L_s$
PNLMS++	$5L_s + (2L_s - 3L_s)/K$
IPNLMS	$6L_s$
GBSG	$3P_s + (2L_s - 3P_s)/K$
SC-IPNLMS	$L_s + 5P_s$
BSC-IPNLMS	$L_s + 5P_s$
Proposed	$5P_s + P_s/K + (2L_s - 5P_s)/K$

TABLE II
COMPUTATIONAL COMPLEXITY MEASURED IN NUMBER OF SUBTRACTIONS, ADDITIONS, COMPARISONS AND SORTING INSTRUCTIONS PER ITERATION.

Algorithm	sub & add	comparisons
NLMS	$2L_s$	0
PNLMS	$3L_s$	L_s
PNLMS++	$3L_s$	L_s
IPNLMS	$4L_s$	0
GBSG	$3P_s$	$(L_s \log_2(L_s) + 2L_s + P_s)/K$
SC-IPNLMS	$L_s + 3P_s$	$(L_s \log_2(L_s))$
BSC-IPNLMS	$L_s + 3P_s$	$(L_s + L_s/B_s \log_2(L_s/B_s))$
Proposed	$3P_s + P_s/K$	$(L_s + L_s/B_s \log_2(L_s/B_s))/K$

C. Complexity analysis

In this paper, the computational complexity is measured by the number of multiplications, additions and comparisons required. For the auxiliary filter NLMS requires $2L_s$ multiplications and $2L_s$ additions (L_s multiplications and L_s additions for calculating the error and similarly for updating the filter). The calculation of $\mathbf{x}_s^T(l)\mathbf{x}_s(l)$ can be neglected if implemented efficiently [20]. The PNLMS algorithm requires $5L_s$ multiplications, $3L_s$ additions and L_s comparisons. The increase in multiplications and additions, compared to the NLMS, comes from the calculation of $\mathbf{x}_s^T(l)\mathbf{G}(l-1)\mathbf{x}_s(l)$. The IPNLMS requires an additional L_s multiplication together with L_s addition for the calculation of $\mathbf{G}(l)$. GBSG uses $3L_s$ multiplications and $3L_s$ additions and the sorting algorithm requires $(L_s \log_2(L_s) + 2L_s + P_s)/K$ sorting instructions when sorting every K :th iteration. The sorting algorithms for the proposed algorithm require only $(L_s + L_s/B_s \log_2(L_s/B_s))/K$ sorting instructions since it is only the blocks that are sorted, i.e. not the individual taps. By only updating $\mathbf{G}(l)$ every K :th iteration, the additional additions and multiplications required can be reduced by a factor K . The complexity of the algorithms in section III are given in table I, which shows the multiplications required, and table II, which shows the additions and comparisons required. Table III shows an example where $L = 2000$, $D = 4$ (giving $L_s = L/D = 500$), $K = 10$, $B_s = 10$ and $P_s = 80$ are used.

It can be seen in table III that the proposed updating scheme has the lowest complexity with 40% of that of the NLMS. Second lowest is the GBSG which has 56% of the NLMS complexity. The main complexity contribution of the GBSG is the sorting. Considering this, it is clear that the block structure used by the proposed algorithms allows a significant complexity reduction since fewer elements are sorted. The most demanding method is the SC-IPNLMS, which is even

TABLE III
COMPUTATIONAL COMPLEXITY MEASURED IN TOTAL NUMBER OF INSTRUCTIONS PER ITERATION, USING THE VALUES $K = 10$, $L_s = 500$, $B_s = 10$ AND $P_s = 80$.

Algorithm	Example
NLMS	2000
PNLMS	4500
PNLMS++	4300
IPNLMS	5000
GBSG	1110
SC-IPNLMS	6040
BC-IPNLMS	1840
Proposed	790

more demanding than a regular IPNLMS. This is due to a full-length sorting every iteration.

D. Simulations

The proposed method for updating the auxiliary filter was evaluated in a simulated environment. In the first simulation setup, flat spectrum noise with zero mean and unit variance was used as input signal $x(n)$ with the sampling frequency $F = 8$ kHz. The output signal was obtained as

$$y(n) = \mathbf{x}(n)^T \mathbf{h}(n) + w(n) \quad (20)$$

where $\mathbf{h}(n)$ is the IR used in the simulation, $w(n)$ is the background noise signal. In the simulations, $w(n)$ is a flat spectrum signal, uncorrelated with $x(n)$ and variance set so that the subsampled echo to the background noise level is -25 dB. An echo path change occurs after 5 s, i.e.

$$\mathbf{h}(n) = \begin{cases} \mathbf{h}_1 & n < 5F \\ \mathbf{h}_2 & n \geq 5F \end{cases}, \quad (21)$$

where \mathbf{h}_1 and \mathbf{h}_2 are two impulse responses shown in figure 4. The length of the IR was 2000 coefficients. The dispersive regions of \mathbf{h}_1 and \mathbf{h}_2 covered 14% of the whole IR. The subsampled signals $x_s(l)$ and $y_s(l)$ were obtained by low-pass filtering and decimation by the factor D . Performance of the adaptive algorithms was evaluated using the normalized misalignment

$$\eta = \frac{\|\hat{\mathbf{h}}_s(l) - \mathbf{h}\|_2}{\|\mathbf{h}\|_2}. \quad (22)$$

In the first simulation setup, the performance of the proposed algorithm for updating the auxiliary filter in comparison with NLMS, IPNLMS, SC-NLMS, BSC-NLMS and GBSG algorithms was evaluated. The parameters in this simulation were set according to the values given in table IV. All algorithms were tuned to yield approximately the same misalignment in order to fairly compare the convergence rate. This was achieved by setting the same step-size parameter μ and tuning the regularization parameters for the different algorithms as shown in table IV. For more details on the regularization tuning, see [12], [17]. The choice of α is as recommended in [12], and the choice of K is as used in [14]. The setting of $B = 40$ (where $B = DB_s$) was chosen to get a good trade-off between convergence speed (low B gives high convergence speed) and computational complexity (high B gives lower computational complexity).

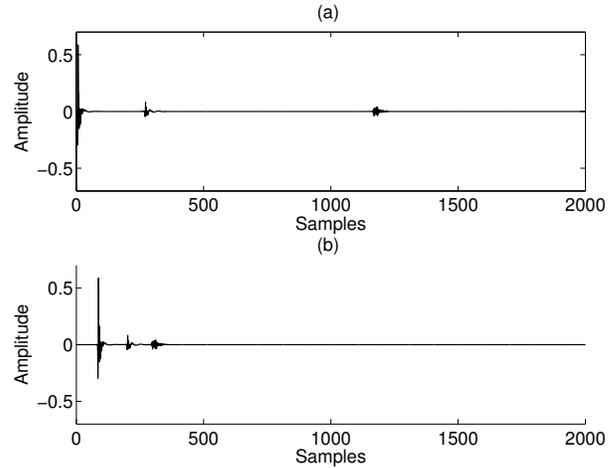


Fig. 4. The impulse responses used in the simulations. It should be noted that the impulse responses \mathbf{h}_1 in (a) and \mathbf{h}_2 in (b) only differ by the location of the tree dispersive regions.

TABLE IV
PARAMETERS USED IN THE SIMULATIONS.

$K = 10$	$L = 2000$	$B = 40$
$D = 4$	$L_s = 500$	$P_s = 80$
$B_s = 10$	$\alpha = -0.5$	$\mu = 0.6$
$\delta_{\text{NLMS}} = 0.2$	$\delta_{\text{IPNLMS}} = \frac{1-\alpha}{2L_s} \delta_{\text{NLMS}} = 0.0003$	$\delta_{\text{GBSG}} = \delta_{\text{IPNLMS}}$
$\delta_{\text{SC-IPNLMS}} = \delta_{\text{IPNLMS}}$	$\delta_{\text{BSC-IPNLMS}} = \delta_{\text{IPNLMS}}$	$\delta_{\text{Prop}} = \delta_{\text{IPNLMS}}$

Figure 5 shows the performance of the proposed algorithm with and without the partial filtering, described by (17). It can be seen that the performance in terms of convergence is maintained while the partial filtering gives a reduced computational complexity.

In figure 6 the performance of the proposed algorithm with and without faster starting convergence, given by (18), and the echo-path change reset function, given by (19), are shown. From figure 6 it can be seen that the faster starting convergence and the echo-path change reset function results in faster initial convergence and that the echo-path change reset function give faster convergence after an echo-path change.

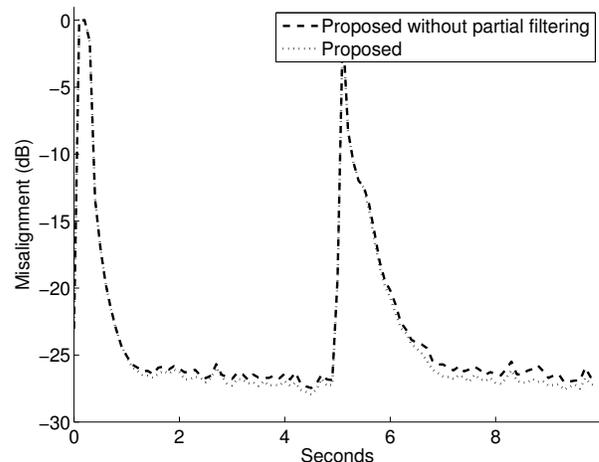


Fig. 5. Filter misalignment for the proposed algorithm with and without partial filtering.

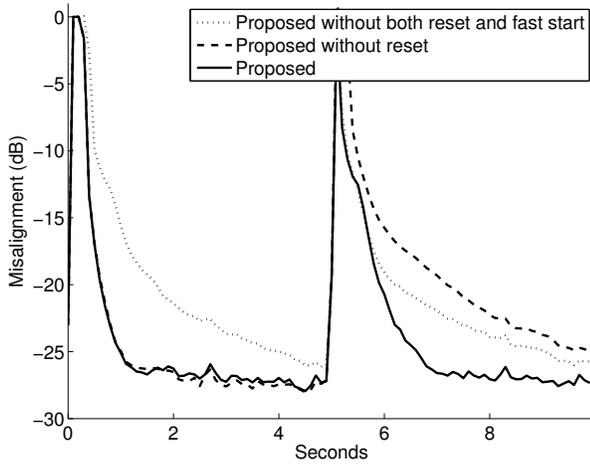


Fig. 6. Misalignment of the filter for the proposed algorithm with and without using faster starting convergence, given by (18), and the echo-path change reset function, given by (19).

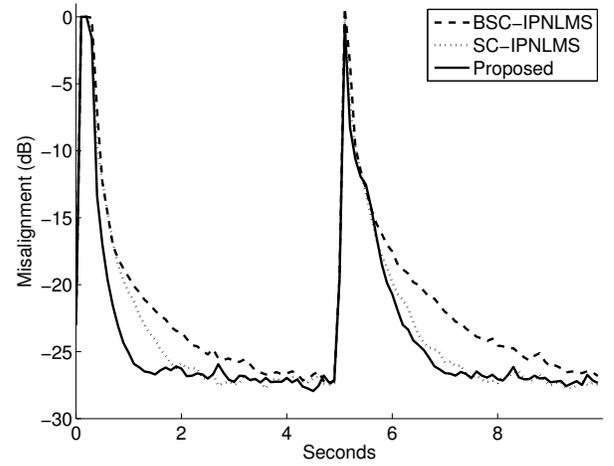


Fig. 8. Misalignment of the filter for the proposed algorithm and SC-IPNLMS, BSC-IPNLMS.

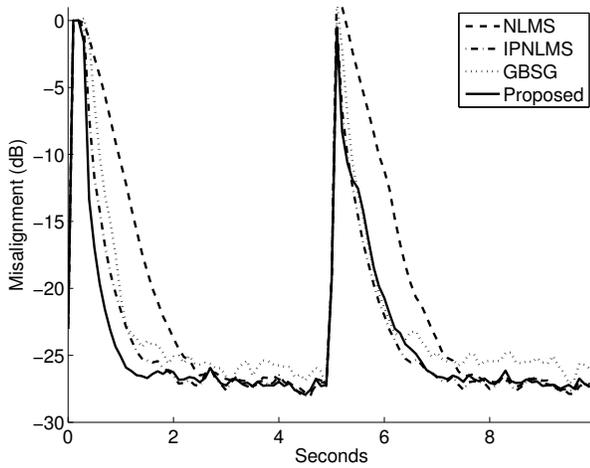


Fig. 7. Misalignment of the filter for the proposed algorithm and NLMS, IPNLMS, GBSG.

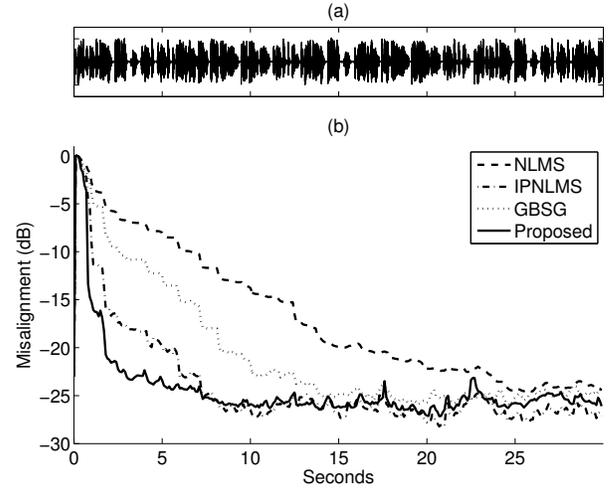


Fig. 9. (a) Speech signal used, (b) Misalignment of the filter for the proposed algorithm and NLMS, IPNLMS, GBSG when using a speech signal and no echo-path change.

The conclusion that can be drawn from the results of the first simulation setup is thus that the partial filtering gives reduced complexity while maintaining the convergence characteristics and that the faster starting convergence and echo-path change reset functions gives improved convergence during startup and echo-path change situations.

In figure 7 the performance of the proposed algorithm is compared with that of the NLMS, IPNLMS and the GBSG algorithms. It can be seen that the performance of the proposed algorithm is comparable to that of the IPNLMS and slightly better than the GBSG. The proposed algorithm was also compared to the SC-IPNLMS and the BSC-IPNLMS which is shown in figure 8 and it can be seen that the proposed algorithm is superior in terms of convergence speed while demanding significantly less overall computational power (see table III).

In the second simulation setup, speech was used as input signal. The output signal was obtained by (20), with the difference that no echo-path change was made, i.e.

$$\mathbf{h}(n) = \mathbf{h}_1. \quad (23)$$

In this simulation the proposed algorithm in relation to NLMS, IPNLMS, SC-NLMS, BSC-NLMS and GBSG were evaluated. All parameters from the previous simulations setup were used, except for δ_{GBSG} , which had to be raised to 4 to yield comparable misalignment of the converged filter to the other algorithms.

When the speech signal is used for adaption of the filters, the convergence time for the proposed algorithm is still faster than IPNLMS, GBSG and NLMS, which can be seen in figure 9. In figure 10 it can also be seen that the proposed algorithm is also faster than both SC-IPNLMS and BSC-IPNLMS.

Thus, the simulations show that even though the proposed algorithm has the lowest complexity it still performs better or comparably in terms of convergence compared to the methods with higher complexity. The full update IPNLMS is outperformed in the speech signal simulation, even though the computational complexity for IPNLMS is more than 5 times as high.

Also, it should be noted that the intended use of the proposed method is to locate the dispersive region which

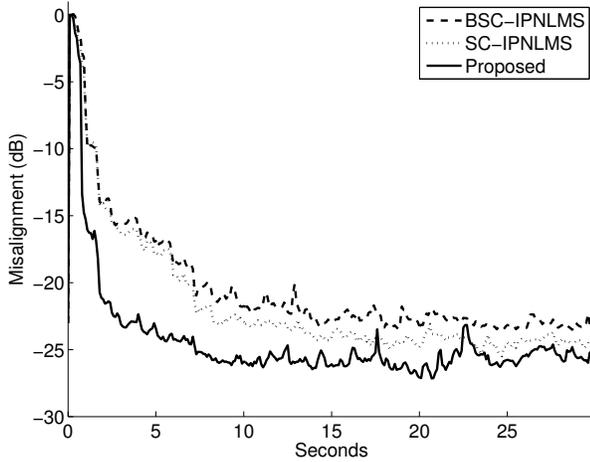


Fig. 10. Misalignment of the filter for the proposed algorithm and SC-IPNLMS, BSC-IPNLMS when using a speech signal and no echo-path change.

only requires the filter to be converged enough so that the coefficient in the dispersive region could be distinguished from the coefficients in the flat delay. This will most likely take place before the filter is fully converged. Therefore the initial convergence speed is the most essential part of the convergence stages.

E. Control of the auxiliary filter

In order to prevent that a DTD reduces the convergence speed, the auxiliary filter is not halted during double-talk. This means that the filter instead runs the risk of divergence during double-talk. However, the dispersive regions are only determined when the filter is said to have converged. This idea is somewhat similar to the two-path echo cancellation approach [25] where a constantly updating background filter is copied to a fixed foreground filter when the background filter is said to have converged.

The filter is said to have converged when the following condition is satisfied

$$\frac{\|\hat{\mathbf{h}}_{s,j}(l) - \hat{\mathbf{h}}_{s,j}(l - T_{\text{AUX}})\|}{\|\hat{\mathbf{h}}_{s,j}(l)\|} < \kappa, \quad (24)$$

where the threshold $\kappa < 1$ determines how stationary the filter must be before it is said to have converged, and T_{AUX} determines the considered time window.

IV. IDENTIFYING DISPERSIVE REGIONS

As stated previously, the single purpose of the auxiliary filter is to locate the dispersive regions. When the filter has converged, the localization of the dispersive regions will be equivalent to finding the subblocks having the most energy, over a certain threshold. The active set S_a gives the P_B largest filter blocks. However, the active blocks do not necessarily perfectly correspond to dispersive regions, although if P_B is large enough, it is clear that all dispersive regions will be within the active blocks. The dispersive region blocks are the blocks $\hat{\mathbf{h}}_{s,j}(l)$, $j \in S_d$, where S_d is the dispersive region set, defined by

$$S_d = \{j_1, j_2, \dots : \|\hat{\mathbf{h}}_{s,j}(l)\|_{\infty} > T_{\text{disp}}, \} \quad (25)$$

where T_{disp} is the dispersive region threshold. The dispersive region set contains a number of subsets,

$$S_d = \{S_{d,1} \cup S_{d,2} \cup \dots \cup S_{d,M}\}, \quad (26)$$

where each subset corresponds to a dispersive region and only contains sequet indexes,

$$S_{d,i} = \{j : j \in S_d, \{j, j+1\} \in S_d \text{ or } \{j, j-1\} \in S_d\}. \quad (27)$$

The delay to the i :th dispersive region can be found by

$$\Delta_i = B \min_{j \in S_{d,i}} \{j\}, \quad (28)$$

and the length of this dispersive region will be

$$L_i = B \left(\max_{j \in S_{d,i}} \{j\} - \min_{j \in S_{d,i}} \{j\} + 1 \right). \quad (29)$$

V. DOUBLE-TALK DETECTION

A typical DTD is defined so that double-talk is said to be present whenever the condition

$$\xi(n) > T_{\text{DTD}} \quad (30)$$

is satisfied. Here $\xi(n)$ is the detection variable and T_{DTD} is a threshold for double-talk detection [26]. The detection variable can depend on a number of given signals and variables, e.g.

$$\xi(n) = \xi(x(n), y(n), \psi, \dots), \quad (31)$$

where ψ is a hold parameter, controlling the number of previous samples to be considered when making the double-talk decision.

In this paper, a time-variable hold parameter $\psi(l)$ is used in the DTD step for improved performance.

A. DTD for the short adaptive filters and NLP

Correct tuning of the parameter $\psi(l)$ is crucial for the performance of the DTD. If $\psi(l)$ is set too small, situations may occur where dispersive regions are out of range and in these situations echo could be detected as double-talk. For the short adaptive filters, this means reduced convergence speed since the adaptation is halted. For the NLP this is even more problematic as double-talk means that the NLP will lessen the attenuation, which in turn could result in audible echoes as there is no masking far-end speech. On the other hand, if $\psi(l)$ is too large, situations may occur where double-talk is not detected, which in turn will cause insufficient duplex due to unnecessary attenuation.

However, since the long auxiliary filter $h_s(n)$ provides information about the IR, it is proposed that this information can be used to set the parameter $\psi(l)$ as

$$\begin{aligned} \psi(l) &= \Delta_{i_{\max}}(l) + L_{i_{\max}}(l), \\ i_{\max} : \Delta_{i_{\max}}(l) &= \max(\Delta_i(l)) \end{aligned} \quad (32)$$

B. Verification of the DTD performance

Two simulation cases were performed to verify the performance of the proposed DTD. It should be noted that the proposed scheme can be used with various DTD algorithms, such as e.g. the classic Geigel detector [18] or normalized cross-correlation [19] (in which $\psi(l)$ could be used to decide the size of the autocorrelation matrix and/or the length of the cross correlation vector), and in this paper, for the sake of simplicity, a modification of the classic Geigel detector

$$\xi_g(n) = \frac{|y(n)|}{\max\{|x(n)|, |x(n-1)|, \dots, |x(n-\psi(l)+1)|\}} \quad (33)$$

is used where a running max replaces the max-operator to reduce the computational complexity.

The first simulation setup was a single-talk setup, where the input signal $x(n)$ constitutes two flat spectrum noise bursts with duration 0.1 seconds and 0.05 seconds, respectively. The signal $y(n)$ was created using (20), with flat spectrum background noise at a level of -35 dB and the filter h_1 in figure 4 (a) with a length of 2000 coefficients. A total number of $M = 2$ short filters, all of length $L_i = 100$, were used to cancel the echo. It should be noted that the actual number of filters required to cancel this echo is 3 filters with lengths 100. The setting $M = 2$ was chosen to simulate a case where the computational power or computer memory could not afford to use sufficient coefficients for the echo cancellation. The resulting smoothed averages of the signals $x(n)$ and $y(n)$, denoted $|x(n)|$ and $|y(n)|$ respectively, are shown in figure 11 (a) and the smoothed average of the echo-cancelled signal $e(n)$, denoted $|e(n)|$, is shown in figure 11 (b).

The second simulation setup was a double-talk situation. Here the same input signal $x(n)$ was used. The output signal $y(n)$ was obtained by filtering $x(n)$ with the filter h_2 , shown in figure 4 (b), and adding a signal $v(n)$ in the form of flat spectrum noise bursts to simulate far-end speech at time interval 0.2 s - 0.3 s and 0.6 s - 0.65 s. A total number of $M = 3$ short filter with lengths $L_i = 100$ were used to cancel the echo in this simulation. Hence the whole echo could therefore be cancelled. The results of the simulations are shown in figure 11 and figure 12. It should be noted that, although appearing similar when comparing figure 11 and figure 12, the two simulation setups are inherently different. In the first setup, the rise of $|e(n)|$ at time interval 0.25 s - 0.35 s and 0.65 s - 0.70 s, is due to the insufficient length of the adaptive echo-cancellation filter, while in the second setup the echo-cancellation filter length is indeed adequate, and the rise of $|e(n)|$ at time interval 0.2 s - 0.3 s and 0.6 s - 0.65 s is due to the far-end noise bursts. Figures 11 and 12 thus illustrate the double-talk detection problem in a line echo-cancellation context.

Different double-talk detection scenarios for the single-talk setup is shown in figure 13. Two different fixed settings of $\psi = \{500, 1500\}$ are compared with the adaptive setting described by (32). It can be seen from figure 13 that both the adaptive setting as well as $\psi = 1500$ work well, i.e. that doubletalk is not detected, while the fixed setting $\psi = 500$ causes the DTD to falsely recognize the echo as far-end activity.

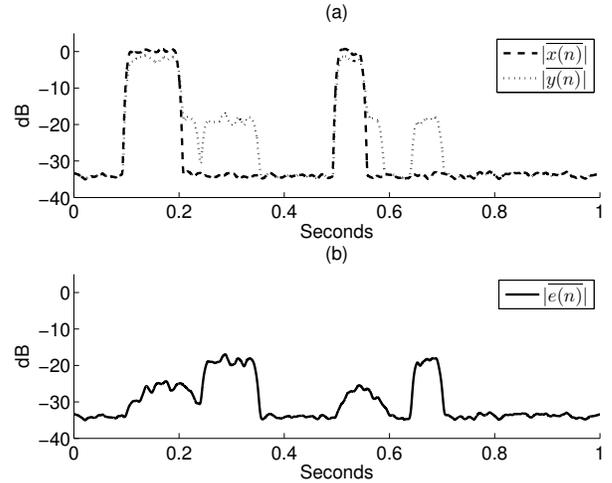


Fig. 11. Single-talk situation with a long impulse response and a too short adaptive echo-cancellation filter. Plot (a) shows smoothed averages of the signals $x(n)$ and $y(n)$, respectively, and plot (b) shows the corresponding smoothed output error.

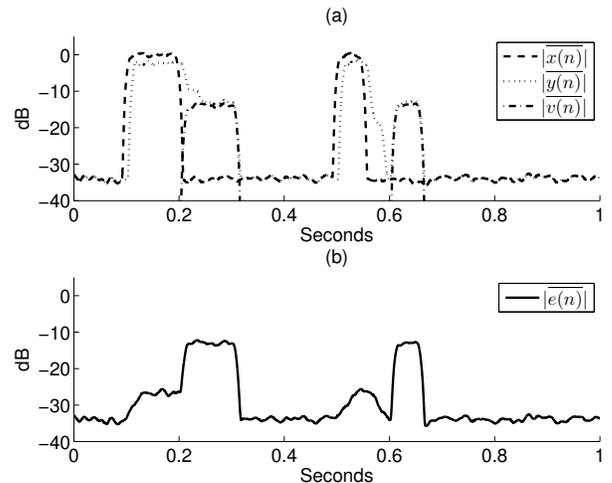


Fig. 12. Double-talk situation with a short impulse response and an echo-cancellation filter of adequate length. Plot (a) shows smoothed averages of the signals $x(n)$ and $y(n)$, respectively, as well as the smoothed average of the far-end disturbance signal $v(n)$. Plot (b) shows the corresponding smoothed output error.

Figure 14 shows the double-talk detection in the double-talk scenario for the same settings of ψ as previously. In this case it is clear that the adaptive setting of ψ as well as the fixed setting $\psi = 500$ both induce the DTD to detect double-talk when far-end activity indeed is present (0.25 s - 0.3 s and 0.6 s - 0.65 s), while the fixed setting of $\psi = 1500$ causes the DTD to miss this detection.

Thus, it is clear that the information from the auxiliary filter stored in the external memory can be used to control the parameter ψ in order to obtain improved DTD performance.

VI. CONCLUSIONS

The paper has presented a solution for network echo cancelling in devices with limited on-chip internal memory and large external memory by combining methods for auxiliary filtering and sparse adaptive filtering with double-talk detec-

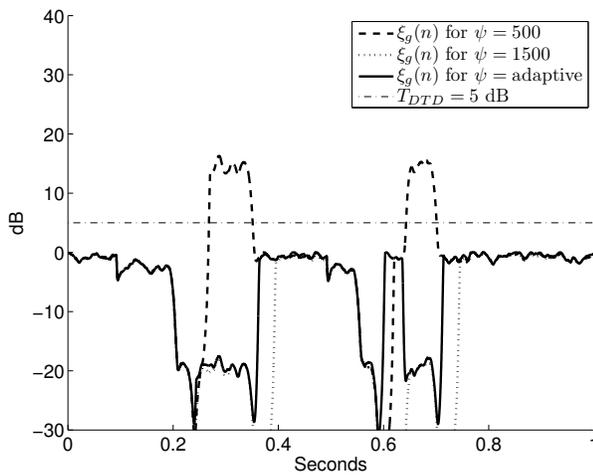


Fig. 13. Single-talk situation with a long impulse response and a too short adaptive echo-cancellation filter. The double-talk detection variable $\xi(n)$ is shown for three different settings of the parameter ψ .

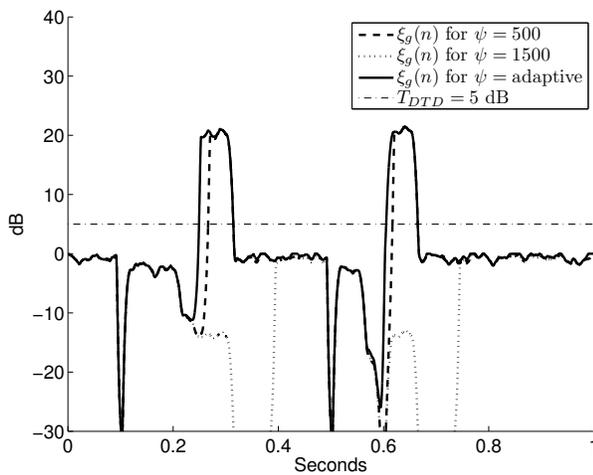


Fig. 14. Double-talk situation with a short impulse response and an echo-cancellation filter of adequate length. The double-talk detection variable $\xi(n)$ is shown for three different settings of the parameter ψ .

tion and non-linear processing. An approach for sparse proportionate adaptive filtering in the described context has been proposed and evaluated through simulations. It has been shown that the proposed solution exhibits similar behavior in terms of convergence speed and steady-state error compared to previous solutions, while requiring less computational resources. Auxiliary filtering and the identification of dispersive regions have been discussed. Finally, a solution for improving double-talk detection by the use of information from the auxiliary filter has been proposed and evaluated through simulations. It has been shown that utilizing the auxiliary filter information can be effective for improving the double-talk detection performance.

REFERENCES

- [1] A. A. Gokhale, *Introduction to telecommunications, 2E*, Thomson Delmar Learning, 2006.
- [2] B. Klepec and A. Kos, "Performance of VoIP application in a simple differentiated service network architecture," in *Proceedings of EUROCON '2001*, 2001, pp. 214–217.
- [3] P. H. Wittke and R. J. Penstone, "Measurements of echo parameters pertinent to high-speed full-duplex data transmission on telephone circuits," *IEEE Journal on Selected Areas in Communication: Analog and Digital Signal Processing*, vol. SAC-2, no. 5, Sept 1984.
- [4] S. Kawamura and M. Hatori, "A tap selection algorithm for adaptive filters," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 1986, number 5, pp. 2979–2982.
- [5] A. Sugiyama, H. Sato, A. Hiano, and S. Ikeda, "A fast convergence algorithm for adaptive FIR filters under computational constraint for adaptive tap-position control," *IEEE Trans. Circuits Syst. II*, vol. 43, pp. 629–636, Sept 1996.
- [6] A. Sugiyama, "A fast convergence algorithm for sparse-tap adaptive fir filters identifying an unknown number of dispersive regions," *IEEE Transactions on Signal Processing*, vol. 50, no. 12, Dec 2002.
- [7] D. L. Duttweiler, "Subsampling to estimate delay with application to echo cancelling," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-31, no. 5, Oct 1983.
- [8] V. A. Margo, D. M. Etter, N. C. Carlson, and J. H. Gross, "Multiple short-length adaptive filters for time-varying echo cancellation," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 1993, pp. 1161–1164.
- [9] P. C. Yip and D. M. Etter, "An adaptive multiple echo canceller for slowly time-varying echo paths," *IEEE Transactions on Communication*, vol. 38, pp. 1693–1698, 1990.
- [10] D. L. Duttweiler, "Proportionate normalized least mean square adaptation in echo cancelers," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 5, pp. 508–518, 2000.
- [11] S. L. Gay, "An efficient, fast converging adaptive filter for network echo cancellation," in *Proceedings of Asilomar Conference on Signals Systems and Computers*, Nov 1998, vol. 1, pp. 394–398.
- [12] J. Benesty and S. L. Gay, "An improved PNLMS algorithm," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '02)*, May 2002, vol. 2, pp. 1881–1884.
- [13] J. Cui, P. A. Naylor, and D. T. Brown, "An improved IPNLMS algorithm for echo cancellation in packet-switched networks," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '04)*, May 2004, vol. 4, pp. 141–144.
- [14] T. Gänslér, J. Benesty, M. M. Sondhi, and S. L. Gay, "Dynamic resource allocation for network echo cancellation," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '01)*, May 2001.
- [15] H. Deng and M. Doroslovacki, "New sparse adaptive algorithms using partial update," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '04)*, May 2004, vol. 2, pp. 845–848.
- [16] K. Dogancay and O. Tanrikulu, "Adaptive filtering algorithms with selective partial updates," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, no. 8, pp. 762–769, 2001.
- [17] J. D. Gordy, T. Aboulnasr, and M. Bouchard, "Reduced-complexity proportionate nlms employing block-based selective coefficient updates," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2008, pp. 233–236.
- [18] D. L. Duttweiler, "A twelve-channel digital echo canceler," *IEEE Transactions on Communications*, vol. COM-26, pp. 647–653, May 1978.
- [19] J. Benesty, D. R. Morgan, and J. H. Cho, "A new class of doubletalk detectors based on cross-correlation," *IEEE Transactions on Speech and Audio Processing*, vol. 8, pp. 168–172, March 2000.
- [20] E. Hänsler and G. Schmidt, *Acoustic Echo and Noise Control: A Practical Approach*, Wiley, 2004.
- [21] S. Haykin, *Adaptive Filter Theory, Forth Edition*, Prentice Hall, 2001.
- [22] J. H. Gross, D. M. Etter, V. A. Margo, and N. C. Carlson, "A block selection adaptive delay filter algorithm for echo cancellation," in *Proceedings of the 35th Midwest Symposium on Circuits and Systems*, Aug 1992, pp. 895–898.
- [23] E. Hänsler and G. Schmidt, *Topics in Acoustic Echo And Noise Control*, Springer-Verlag Berlin Heidelberg, 2006.
- [24] O. Tanrikulu and K. Dogancay, "Selective-partial-update proportionate normalized least-mean-squares algorithm for network echo cancellation," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2002, vol. 2, pp. 1889–1892.
- [25] F. Lindstrom, C. Schüldt, and I. Claesson, "An improvement of the two-path algorithm transfer logic for acoustic echo cancellation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, pp. 1320–1326, May 2007.
- [26] S. L. Gay and J. Benesty, *Acoustic Signal Processing for Telecommunication*, Kluwer Academic Publishers, 2000.



Magnus Berggren was born in Skellefteå, Sweden. He received his M.Sc. degree in engineering physics from Umeå University, Umeå, Sweden, in 2008. Currently he is pursuing his Ph.D. studies at the Department of Electrical Engineering, Blekinge Institute of Technology, Karlskrona, Sweden. His research interests include speech and audio processing.



Ingvar Claesson (M'91) received the M.Sc. degree in 1980 and the Ph. D. degree in 1986 in Electrical Engineering at University of Lund, Sweden. He was appointed Senior Lecturer in Telecommunication Theory at Lund University in 1986, and was appointed Associate Professor in 1992. In 1990, he was one of the founders of the Department of Signal Processing, Blekinge Institute of Technology, and is currently Head of Research and Principal Supervisor in Signal Processing. Since 1998, he holds the chair of Applied Signal Processing at Blekinge Institute of Technology, Karlskrona, Sweden, and also served as Research Dean 2005-2011. His current research interests are in adaptive signal processing, blind equalization, adaptive beamforming, speech enhancement, blind signal separation, active noise control, health applications, filter design and remote laboratories. A keen interest for applications has lead to more than 20 patents and he regularly serves as industry consultant.



Markus Borgh was born in Sollefteå, Sweden. He received his M.Sc. degree in engineering physics from Umeå University, Umeå, Sweden, in 2008. Currently he is with Limes Audio AB, Umeå, Sweden and pursuing his Ph.D. studies. His research areas are noise reduction, speech enhancement and echo cancelation.



Christian Schüldt was born in Stockholm, Sweden. He received his B.Sc. degree in electrical engineering with specialization in mobile communication from the Royal Institute of Technology, Stockholm, Sweden, in 2001 and his M.Sc. degree in electrical engineering with specialization in signal processing and computer vision from the Royal Institute of Technology, Stockholm, Sweden, in 2004. Currently he is pursuing his Ph.D. studies at the Department of Electrical Engineering, Blekinge Institute of Technology, Karlskrona, Sweden. His areas of

professional interest include signal processing techniques and real-time programming.



Fredric Lindström was born in Skellefteå, Sweden. He received his M.Sc. degree in applied physics from Uppsala University, Uppsala, Sweden, in 2001 and the Ph. D. degree from Blekinge Institute of Technology, Ronneby, Sweden, in 2007. Since 2007, he is the CEO of Limes Audio AB, Umeå, Sweden. His current research interest is adaptive signal processing with applications in hands-free systems, e.g. acoustic echo canceling, acoustic echo suppression techniques, and algorithms for finite precision implementations.