Hindawi Publishing Corporation
EURASIP Journal on Audio, Speech, and Music Processing
Volume 2008, Article ID 156960, 9 pages
doi:10.1155/2008/156960

Research Article

Frequency-Domain Adaptive Algorithm for Network Echo Cancellation in VoIP

Xiang (Shawn) Lin,[1] Andy W. H. Khong,[1] Miloš Doroslovački,[2] and Patrick A. Naylor[1]

[1] *Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, UK*
[2] *Department of Electrical and Computer Engineering, The George Washington University, Washington, DC 20052, USA*

Correspondence should be addressed to Xiang (Shawn) Lin, shawn.lin04@imperial.ac.uk

Received 1 November 2007; Accepted 8 April 2008

Recommended by Sen Kuo

We propose a new low complexity, low delay, and fast converging frequency-domain adaptive algorithm for network echo cancellation in VoIP exploiting MMax and sparse partial (SP) tap-selection criteria in the frequency domain. We incorporate these tap-selection techniques into the multidelay filtering (MDF) algorithm in order to mitigate the delay inherent in frequency-domain algorithms. We illustrate two such approaches and discuss their tradeoff between convergence performance and computational complexity. Simulation results show an improvement in convergence rate for the proposed algorithm over MDF and significantly reduced complexity. The proposed algorithm achieves a convergence performance close to that of the recently proposed, but substantially more complex improved proportionate MDF (IPMDF) algorithm.

1. INTRODUCTION

The popularity of voice over internet protocol (VoIP) coupled with an increasing expectation for natural communication over packet-switched networks has called for improvement in VoIP technologies in recent years. As network systems migrate from traditional voice telephony over public switch telephone network (PSTN) to packet-switched networks for VoIP, improving the quality of services (QoS) for VoIP has been and will remain a challenge [1, 2]. As described in [1], several factors that can affect the QoS for VoIP include the choice of speech coder-decoders (codecs) [3], algorithmic processing delay [4], and packet loss [5], where the algorithmic delay is one of the significant factors for determining the budget for delay introduced by network echo cancellers. The problem of network echo is introduced by the impedance mismatch between the 2- and 4-wire circuits of a network hybrid [6], which occurs in VoIP systems, where analog phones are involved in PC-to-phone or phone-to-phone connections [7], where "PC" represents all-digital terminals. Acoustic echo, on the other hand, occurs when hands-free conversations are conducted [8]. Transmission and algorithmic processing cause the echo to be transmitted back to the originator with a delay, hence impeding effective communication. As a result, network echo cancellation for IP networks has received increased attention in recent years. For effective network echo cancellation (NEC), adaptive filters such as shown in Figure 1 have been employed for the estimation of network impulse response. Using the estimated impulse response, a replica of the echo is generated and subtracted from the far-end transmitted signal. The main aim of this work is therefore to address the problem of (NEC) with reduced complexity and low algorithmic delay through the use of adaptive algorithms.

In VoIP systems, where traditional telephony equipment is connected to the packet-switched network, the resulting network impulse response such as shown in Figure 2 is typically of length 64–128 milliseconds. This impulse response exhibits an "active" region in the range of only 8–12 milliseconds duration, and, consequently, it is dominated by "inactive" regions, where magnitudes are close to zero making the impulse response sparse. The "inactive" region is principally due to the presence of bulk delay caused by unknown network propagation, encoding, and jitter buffer delays [7]. One of the first algorithms which exploits this sparse nature for the identification of network impulse

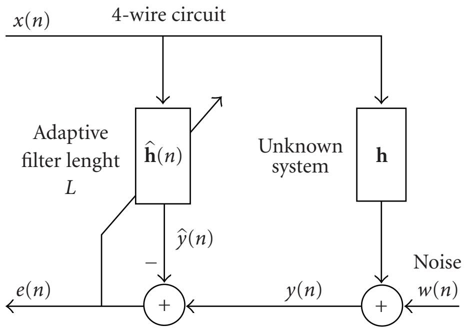


Figure 1: Network echo cancellation.

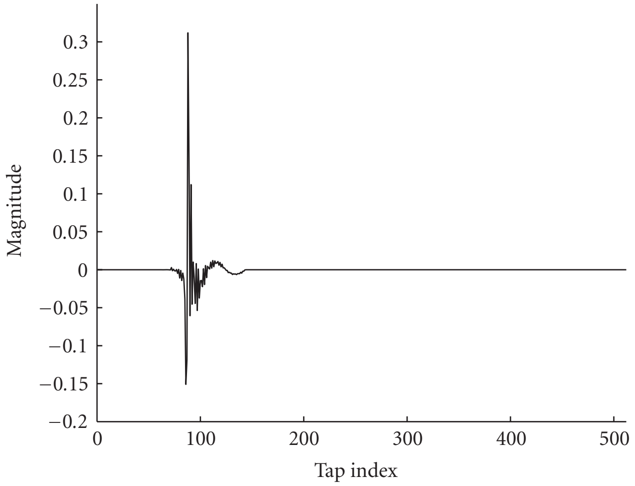


Figure 2: A sparse network echo impulse response, sampled at 8 kHz.

responses is the proportionate normalized least-mean-square (PNLMS) algorithm [9], where each filter coefficient is updated with a step-size which is proportional to the coefficient magnitudes. The PNLMS algorithm is then shown to outperform classical adaptive algorithms with a uniform step-size across all filter coefficients such as the normalized least-mean-square (NLMS) algorithm for NEC application [9]. Although the PNLMS algorithm achieves fast initial convergence, its rate of convergence reduces significantly. This is due to the slow convergence of filter coefficients having small magnitudes. To mitigate this problem, subsequent improved versions such as the improved PNLMS (IPNLMS) [10] and the improved IPNLMS [11] algorithms were proposed. These algorithms share the same characteristic of introducing a controlled mixture of proportionate (PNLMS) and nonproportionate (NLMS) adaptation. Consequently, these algorithms perform better than PNLMS for sparse impulse responses.

The increase in VoIP traffic in recent years has resulted a high demand for high density NEC in which it is desirable to run several hundred echo cancellers in one processor core. Defining L as the length of the impulse response, the PNLMS and IPNLMS algorithms require approximately $\mathcal{O}(3L)$ and $\mathcal{O}(4L)$ number of multiplications per sample iteration respectively compared to $\mathcal{O}(2L)$ for the substantially slower converging NLMS algorithm. Hence, in order to reduce the computational complexity of PNLMS and IPNLMS, the sparse partial update NLMS (SPNLMS) algorithm was recently proposed [12], which combines two adaptation strategies: sparse adaptation for improving rate of convergence and partial-updating for complexity reduction. For the majority of adapting iterations, under the sparse partial (SP) adaptation, only those taps corresponding to tap-inputs and filter coefficients both having large magnitudes are updated. However, from time to time the algorithm gives equal opportunity for the coefficients with smaller magnitude to be updated by employing MMax tap-selection [13]. This only updates those filter taps corresponding to the $M < L$ largest magnitude tap-inputs. It is noted that partial update strategies have also been applied to the filtered-X LMS (FxLMS) algorithms as described in [14, 15]. Other ways to reduce the complexity of adaptive filtering algorithm include the use of a shorter adaptive filter to model only the active region of the sparse impulse responses as described in [16].

It is well known that frequency-domain adaptive filtering such as the fast-LMS (FLMS) algorithm [17] offers an attractive means of achieving efficient implementation. In contrast to time-domain adaptive filtering algorithms, frequency-domain adaptive algorithms incorporate block updating strategies, whereby the fast-Fourier transform (FFT) algorithm [18] is used together with the overlap-save method [19, 20]. However, one of the main drawbacks of these frequency-domain approaches is the delay introduced between the input and output, which is generally equal to the length of the adaptive filter. Since reducing the algorithmic processing delay for VoIP applications is crucial, frequency-domain adaptive algorithms with low delay are desirable especially for the identification of long network impulse responses. The multidelay filtering (MDF) algorithm [21] has been proposed in the context of acoustic echo cancellation for mitigating the problem of delay. This algorithm partitions an adaptive filter of length L into K blocks each of length N. As a result, the delay of MDF algorithm is reduced by a factor of K compared to FLMS. The benefit of low delay for MDF over FLMS in the context of NEC has been shown in [22].

The aim of this work is to develop a *low complexity*, *low delay*, and *fast converging* adaptive algorithm for identifying sparse impulse responses presented in the problem of NEC for VoIP applications. We achieve this by incorporating the MMax and SP tap-selection into the frequency-domain MDF structure. As will be shown in this work, applying the MMax and SP tap-selection to frequency-domain adaptive filtering presents significant challenges since the time-domain sparse impulse response is not necessarily sparse in the frequency domain. We first review in Section 2 the SPNLMS and MDF algorithms. We then propose, in Section 3.1, to incorporate MMax tap-selection into MDF structure for complexity reduction. We show how this can be achieved using two approaches and we compare their tradeoffs in terms of complexity and performance. We next illustrate, in Section 3.2, how the sparseness of the Fourier transformed impulse response varies with the number of blocks K in the MDF structure. Utilizing these results, we show how the SP tap-selection can be incorporated into the MDF structure for fast convergence and low delay. The computational complexity for the proposed algorithm is discussed in Section 3.3. In Section 4, we present the simulation results and discussions using both colored Gaussian noise (CGN) and speech inputs for NEC. Finally, conclusions are drawn in Section 5.

2. REVIEW OF THE SPNLMS AND MDF ALGORITHMS

We first review the problem of sparse system identification. With reference to Figure 1, we define tap-input vector $\mathbf{x}(n)$, network impulse response $\mathbf{h}$, and coefficients of adaptive filter $\widehat{\mathbf{h}}(n)$ as

$$\begin{aligned}\mathbf{x}(n) &= [x(n)\cdots x(n-L+1)]^T,\\ \mathbf{h} &= [h_0\cdots h_{L-1}]^T,\\ \widehat{\mathbf{h}}(n) &= [\widehat{h}_0(n)\cdots \widehat{h}_{L-1}(n)]^T,\end{aligned} \tag{1}$$

where L is the length of $\mathbf{h}$ and $[\cdot]^T$ is defined as vector/matrix transposition. The adaptive filter $\widehat{\mathbf{h}}(n)$, which is chosen to be of the same length as $\mathbf{h}$, will model the unknown impulse response $\mathbf{h}$ using the near-end signal

$$y(n) = \mathbf{x}^T(n)\mathbf{h} + w(n), \tag{2}$$

where $w(n)$ is the additive noise.

2.1. The SPNLMS algorithm

The sparse partial (SP) update NLMS (SPNLMS) algorithm [12] utilizes the sparse nature of network impulse response. This algorithm incorporates two updating strategies: MMax tap-selection [13] for complexity reduction and SP adaptation for fast convergence. Although it is normal to expect that adapting filter coefficients using partial-updating strategies suffers from degradation in convergence performance, it was shown in [12] that such degradation can be offset by the SP tap-selection.

The updating equation for SPNLMS is given by

$$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \mu\frac{\mathbf{Q}(n)\mathbf{x}(n)e(n)}{\|\mathbf{Q}(n)\mathbf{x}(n)\|_2^2+\delta}, \tag{3}$$

where μ is the step-size, δ is the regularization parameter and $\|\cdot\|_2$ is defined as the l_2-norm. As shown in Figure 1, the a priori error is given by

$$e(n) = y(n) - \mathbf{x}^T(n)\widehat{\mathbf{h}}(n-1). \tag{4}$$

The $L\times L$ tap-selection matrix

$$\mathbf{Q}(n) = \operatorname{diag}\{q_0(n)\cdots q_{L-1}(n)\} \tag{5}$$

in (3) determines the step-size gain for each filter coefficient and is dependent on the MMax and SP updating strategies for SPNLMS. The relative significance of these strategies is controlled by the variable $T\in\mathbb{Z}^+$ such that for $\operatorname{mod}(n,T)=0$, elements $q_i(n)$ for $i=0,\ldots,L-1$ are given by

$$q_i(n) = \begin{cases}1 & i\in\{\text{indices of the } M_1 \text{ maxima of } |x(n-i)|\},\\ 0 & \text{otherwise,}\end{cases} \tag{6}$$

and for $\operatorname{mod}(n,T)\neq 0$,

$$q_i(n) = \begin{cases}1 & i\in\{\text{indices of the } M_2 \text{ maxima of}\\ & \qquad |x(n-i)\widehat{h}_i(n-1)|\},\\ 0 & \text{otherwise.}\end{cases} \tag{7}$$

The variables M_1 and M_2 define the number of selected taps for MMax and SP, respectively, and the MMax tap-selection criteria given by (6) for the time-domain is achieved by sorting $\mathbf{x}(n)$ using, for example, the SORTLINE [23] and short sort [24] routines. It has been shown in [12] that, including the modest overhead for such sorting operations, the SPNLMS algorithm achieves lower complexity than NLMS. To summarize, SPNLMS incorporates MMax tap-selection given by (6) and SP tap-selection given by (7) for complexity reduction and fast convergence, respectively.

2.2. The MDF algorithm

The MDF algorithm [21] mitigates the problem of delay inherent in FLMS [17] by partitioning the adaptive filter into K subfilters each of length N, with $L = KN$ and $K\in\mathbb{Z}^+$. As a consequence of this partitioning, the delay for the MDF is reduced by a factor of K compared to FLMS. To describe the MDF algorithm, we define m as the frame index and the following time-domain quantities given by

$$\mathbf{X}(m) = [\mathbf{x}(mN)\cdots\mathbf{x}(mN+N-1)], \tag{8}$$

$$\mathbf{y}(m) = [y(mN)\cdots y(mN+N-1)]^T, \tag{9}$$

$$\widehat{\mathbf{h}}(m) = [\widehat{\mathbf{h}}_0^T(m)\cdots\widehat{\mathbf{h}}_{K-1}^T(m)]^T, \tag{10}$$

$$\begin{aligned}\widehat{\mathbf{y}}(m) &= [\widehat{y}(mN)\cdots\widehat{y}(mN+N-1)]^T\\ &= \mathbf{X}^T(m)\widehat{\mathbf{h}}(m),\end{aligned} \tag{11}$$

$$\mathbf{e}(m) = \mathbf{y}(m) - \widehat{\mathbf{y}}(m). \tag{12}$$

We also define a $2N\times 1$ tap-input vector

$$\boldsymbol{\chi}(m-k) = [x(mN-kN-N)\cdots x(mN-kN+N-1)]^T, \tag{13}$$

where $k=0,\ldots,K-1$ is defined as the block index and the subfilters in (10) are given as

$$\widehat{\mathbf{h}}_k(m) = [\widehat{h}_{kN}(m)\cdots\widehat{h}_{kN+N-1}(m)]^T. \tag{14}$$

We next define $\mathbf{F}_{2N}$ as the $2N\times 2N$ Fourier matrix and a $2N\times 2N$ matrix

$$\underline{\mathbf{D}}(m-k) = \operatorname{diag}\{\mathbf{F}_{2N}\boldsymbol{\chi}(m-k)\} = \operatorname{diag}\{\underline{\boldsymbol{\chi}}(m-k)\}, \tag{15}$$

with diagonal elements containing the Fourier transform of $\boldsymbol{\chi}(m-k)$ for the kth block. We also define the following frequency-domain quantities [8]

$$\begin{gathered}\underline{\mathbf{y}}(m)=\mathbf{F}_{2N}\begin{bmatrix}\mathbf{0}_{N\times 1}\\ \mathbf{y}(m)\end{bmatrix},\qquad \hat{\underline{\mathbf{h}}}_k(m)=\mathbf{F}_{2N}\begin{bmatrix}\hat{\mathbf{h}}_k(m)\\ \mathbf{0}_{N\times 1}\end{bmatrix},\\ \underline{\mathbf{e}}(m)=\mathbf{F}_{2N}\begin{bmatrix}\mathbf{0}_{N\times 1}\\ \mathbf{e}(m)\end{bmatrix},\\ \mathbf{G}^{01}=\mathbf{F}_{2N}\mathbf{W}^{01}\mathbf{F}_{2N}^{-1},\quad \mathbf{W}^{01}=\begin{bmatrix}\mathbf{0}_{N\times N} & \mathbf{0}_{N\times N}\\ \mathbf{0}_{N\times N} & \mathbf{I}_{N\times N}\end{bmatrix},\\ \mathbf{G}^{10}=\mathbf{F}_{2N}\mathbf{W}^{10}\mathbf{F}_{2N}^{-1},\quad \mathbf{W}^{10}=\begin{bmatrix}\mathbf{I}_{N\times N} & \mathbf{0}_{N\times N}\\ \mathbf{0}_{N\times N} & \mathbf{0}_{N\times N}\end{bmatrix},\end{gathered}\tag{16}$$

where $\mathbf{0}_{N\times N}$ is the $N\times N$ null matrix and $\mathbf{I}_{N\times N}$ is the $N\times N$ identity matrix. The MDF algorithm is then given by [21]

$$\underline{\mathbf{e}}(m)=\underline{\mathbf{y}}(m)-\mathbf{G}^{01}\sum_{k=0}^{K-1}\underline{\mathbf{D}}(m-k)\hat{\underline{\mathbf{h}}}_k(m-1),\tag{17}$$

$$\mathbf{S}(m)=\lambda\mathbf{S}(m-1)+(1-\lambda)\underline{\mathbf{D}}^{*}(m)\underline{\mathbf{D}}(m),\tag{18}$$

$$\mathbf{P}(m)=\mathbf{S}(m)+\delta\mathbf{I}_{2N\times 2N}=\operatorname{diag}\{p_0(m)\cdots p_{2L-1}(m)\},\tag{19}$$

$$\hat{\underline{\mathbf{h}}}_k(m)=\hat{\underline{\mathbf{h}}}_k(m-1)+\mu\mathbf{G}^{10}\underline{\mathbf{D}}^{*}(m-k)\mathbf{P}^{-1}(m)\underline{\mathbf{e}}(m),\tag{20}$$

where $*$ denotes complex conjugate, $0\ll\lambda<1$ is the forgetting factor and $\mu=\beta(1-\lambda)$ is the step-size with $0<\beta\leq 1$ [21]. Letting σ_x^2 be the input signal variance, the initial regularization parameters [8] are $\mathbf{S}(0)=\sigma_x^2/100$ and $\delta=20\sigma_x^2N/L$. For $N=L$ and $K=1$, MDF is equivalent to FLMS [17].

3. THE SPARSE PARTIAL UPDATE MULTIDELAY FILTERING ALGORITHM

Our aim is to utilize the low delay inherent in MDF as well as the fast convergence and reduced complexity brought about by combining SP and MMax tap-selection for NEC. We achieve this aim by first describing how MMax tap-selection given in (6) can be incorporated into MDF. We next show, using an illustrative example, how the sparse nature of the impulse response is exploited in the frequency domain which then allows us to integrate the SP tap-selection given by (7). The proposed MMax-MDF and SPMMax-MDF algorithms are described by (17), (18), (19), and

$$\hat{\underline{\mathbf{h}}}_k(m)=\hat{\underline{\mathbf{h}}}_k(m-1)+\mu\mathbf{G}^{10}\tilde{\underline{\mathbf{D}}}^{*}(m-k)\mathbf{P}^{-1}(m)\underline{\mathbf{e}}(m).\tag{21}$$

The difference between (20) and (21) is that the latter employs $\tilde{\underline{\mathbf{D}}}^{*}(m-k)$, and we will describe in the following how this $2N\times 2N$ diagonal matrix can be obtained for the cases of MMax and SP tap-selection criterion.

3.1. The MMax-MDF algorithm

As described in Section 2.1, the MMax tap-selection given in (6) is achieved by sorting $\mathbf{x}(n)$. In the frequency-domain MDF implementation, however, elements in $\tilde{\underline{\mathbf{D}}}(m-k)$ are normalized by elements $p_i(m)$ in the vector $\mathbf{P}(m)$ defined in (19). Hence, for the frequency-domain MMax tap-selection, we select taps corresponding to the M_1 maxima of the Fourier transformed tap-inputs normalized by $p_i(m)$ with $i=0,\ldots,2L-1$. For this tap-selection strategy, the concatenated Fourier transformed tap-input across all K blocks is given as

$$\begin{aligned}\underline{\mathbf{g}}(m)&=\left[\underline{\boldsymbol{\chi}}^T(m)\cdots\underline{\boldsymbol{\chi}}^T(m-K+1)\right]^T\\ &=\left[\underline{\chi}_0(m)\cdots\underline{\chi}_{2L-1}(m)\right]^T,\end{aligned}\tag{22}$$

where $\underline{\boldsymbol{\chi}}(m-k)$ is defined in (15) and $\underline{\chi}_i(m),\ i=0,\ldots,2L-1$ denotes the ith element of $\underline{\mathbf{g}}(m)$. Elements of the $2L\times 2L$ diagonal MMax tap-selection matrix $\mathbf{Q}(m)$ are given by

$$q_i(m)=\begin{cases}1 & i\in\left\{\text{indices of the } M_1 \text{ maxima of } \dfrac{\underline{\chi}_i^{*}(m)\underline{\chi}_i(m)}{p_i(m)}\right\},\\ 0 & \text{otherwise},\end{cases}\tag{23}$$

for $i=0,\ldots,2L-1$ with $1\leq M_1\leq 2L$. Due to the normalization by $p_i(m)$ in (23), we denote this algorithm as MMax-MDF$_{\mathrm{N}}$ and define a $2L\times 1$ vector $\tilde{\underline{\mathbf{g}}}(m)$ containing the subselected Fourier transformed tap-inputs as

$$\tilde{\underline{\mathbf{g}}}(m)=\mathbf{Q}(m)\underline{\mathbf{g}}(m)=\left[\tilde{\underline{\chi}}_0(m)\cdots\tilde{\underline{\chi}}_{2L-1}(m)\right]^T.\tag{24}$$

The $2N\times 2N$ diagonal matrix $\tilde{\underline{\mathbf{D}}}(m-k)$ for MMax-MDF$_{\mathrm{N}}$ is then given by

$$\begin{aligned}\tilde{\underline{\mathbf{D}}}(m-k)=\operatorname{diag}\{&\tilde{\underline{\chi}}_{2kN}(m)\cdots\tilde{\underline{\chi}}_{2kN+2N-1}(m)\},\\ &k=0,\ldots,K-1.\end{aligned}\tag{25}$$

Hence, it can be seen that elements in the vector $\tilde{\underline{\mathbf{D}}}(m-k)$ are obtained from the kth block of the selected Fourier transformed tap-inputs contained in $\tilde{\underline{\mathbf{g}}}(m)$ with indices from $2kN$ to $2kN+2N-1$. The adaptation of MMax-MDF$_{\mathrm{N}}$ algorithm is described by (23)–(25) and (21).

It is noted that the MMax-MDF$_{\mathrm{N}}$ algorithm requires $2L$ additional divisions for tap-selection due to the normalization by $p_i(m)$ in (23). Hence, to reduce the complexity even further, we consider an alternative approach where such normalization is removed so that elements of the $2L\times 2L$ diagonal tap-selection matrix $\mathbf{Q}(m)$ are expressed as

$$q_i(m)=\begin{cases}1, & i\in\{\text{indices of the } M_1 \text{ maxima of } |\underline{\chi}_i(m)|\},\\ 0, & \text{otherwise},\end{cases}\tag{26}$$

for $i=0,\ldots,2L-1$ and $1\leq M_1\leq 2L$. As opposed to MMax-MDF$_{\mathrm{N}}$, we denote this scheme as the MMax-MDF algorithm since normalization by $p_i(m)$ is removed. Accordingly, elements in $\tilde{\underline{\mathbf{D}}}(m-k)$ for MMax-MDF are computed using (24) and (25), where $\mathbf{Q}(m)$ is obtained from (26). Hence, the adaptation of MMax-MDF algorithm is described by (24)–(26) and (21).

As will be shown in Section 4, the degradation in convergence performance due to tap-selection is less in MMax-MDF$_\mathrm{N}$ than in MMax-MDF. However, since reducing complexity is our main concern, we choose to use MMax-MDF as our basis for reducing the computational complexity of the proposed algorithm. As will be described in Section 3.2, the proposed algorithm incorporates the SP tap-selection to achieve, in addition, a fast rate of convergence.

3.2. The SPMMax-MDF algorithm

We show in this section how the SP tap-selection can be incorporated into the frequency domain. The SP tap-selection defined by (7) was proposed to achieve fast convergence for the identification of sparse impulse responses. We note that the direct implementation of SP tap-selection into frequency-domain adaptive filtering such as FLMS is inappropriate since impulse response in the transformed domain is not necessarily sparse. To illustrate this, we study the effect of $K \geq 1$ on the concatenated impulse response of the MDF structure $\underline{\mathbf{h}}$ defined by

$$\underline{\mathbf{h}} = \mathbb{F}_{2L}\left[\left[\begin{matrix}\mathbf{h}_0\\ \mathbf{0}_{N\times1}\end{matrix}\right]^T \cdots \left[\begin{matrix}\mathbf{h}_{K-1}\\ \mathbf{0}_{N\times1}\end{matrix}\right]^T\right]^T, \tag{27}$$

where

$$\mathbf{h}_k = \left[h_{kN} \cdots h_{kN+N-1}\right]^T, \tag{28}$$

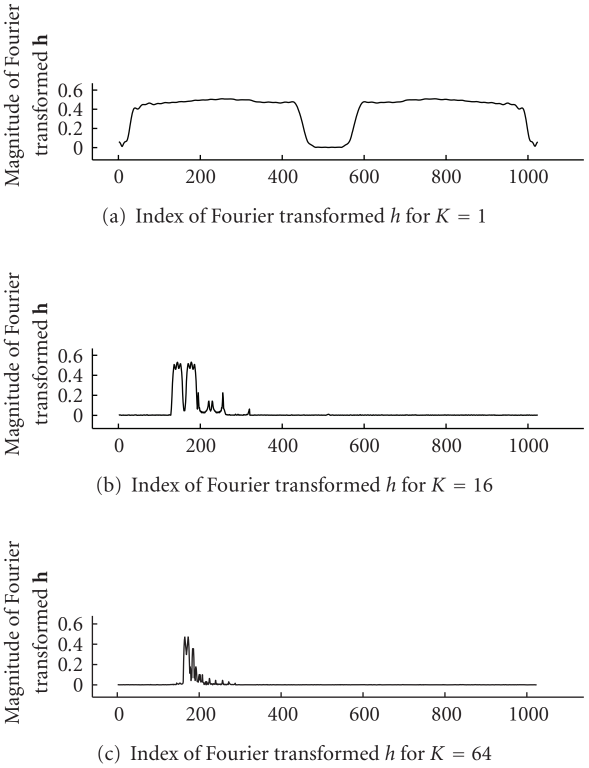
for $k = 0, \ldots, K-1$ is the kth subfilter to be identified and

$$\mathbb{F}_{2L} = \begin{bmatrix}\mathbf{F}_{2N} & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & \mathbf{F}_{2N}\end{bmatrix}_{2L\times2L} \tag{29}$$

is a $2L \times 2L$ matrix constructed by K Fourier matrices each of size $2N \times 2N$. As indicated in (28), the impulse response $\mathbf{h}$ is partitioned into smaller blocks in the time domain as K increases. Figure 3 shows the variation of the magnitude of $\underline{\mathbf{h}}$ for $K = 1$, $K = 16$ and $K = 64$, where MDF is equivalent to FLMS for $K = 1$. As can be seen from the figure, the magnitude of $\underline{\mathbf{h}}$ is not sparse for $K = 1$. Hence SP tap-selection in the MDF structure will not improve the convergence performance for $K = 1$. For the cases where $K > 1$, the number of taps with small magnitudes in $\underline{\mathbf{h}}$ increases with K, that is, the number of subfilters. In Figure 4, we show how the sparseness of the magnitude of $\underline{\mathbf{h}}$ varies with K using the sparseness measure given by [25, 26]

$$\xi = \frac{L}{L-\sqrt{L}}\left[1 - \frac{\|\underline{\mathbf{h}}\|_1}{\sqrt{L}\|\underline{\mathbf{h}}\|_2}\right], \tag{30}$$

where $\|\cdot\|_1$ denotes l_1-norm and it was shown in [26, 27] that ξ increases with the sparseness of $\mathbf{h}$, where $0 \leq \xi \leq 1$. As can be seen from Figure 4, the magnitude of $\underline{\mathbf{h}}$ becomes more sparse as K increases. As a consequence, we would expect SP tap-selection to improve the convergence rate of MDF for sparse system identification.



(a) Index of Fourier transformed h for $K = 1$

(b) Index of Fourier transformed h for $K = 16$

(c) Index of Fourier transformed h for $K = 64$

Figure 3: Variation of the magnitude of $\underline{\mathbf{h}}$ of length $2L$ with $L = 512$ for (a) $K = 1$, (b) $K = 16$, and (c) $K = 64$.

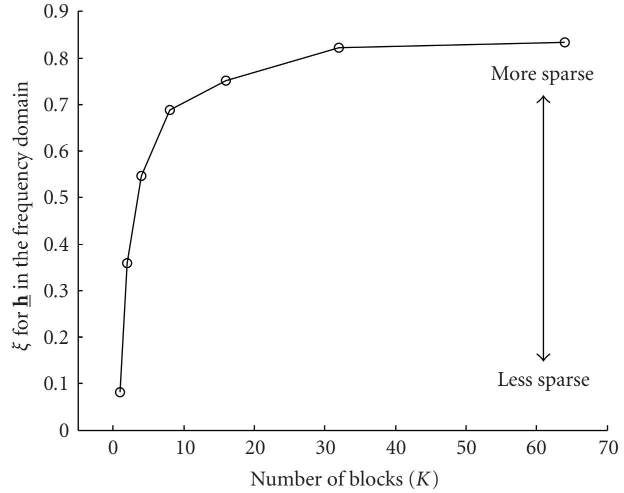


Figure 4: Sparseness of the magnitude of $\underline{\mathbf{h}}$ against K.

Although integrating SP tap-selection can be beneficial in the frequency domain, it requires careful consideration since as can be seen from (13), the length of the input frame $\boldsymbol{\chi}(m-k)$ is $2N$ compared to L for the adaptive filter. This causes a length mismatch between $\underline{\boldsymbol{\chi}}(m-k)$ and $\widehat{\underline{\mathbf{h}}}(m)$. We overcome this problem by concatenating all frequency-domain subfilters, $\underline{\hat{\mathbf{h}}}_k^T(m)$, $k = 0, \dots, K-1$ to obtain $\underline{\hat{\mathbf{h}}}(m)$, which is of length $2L$, that is,

$$\begin{aligned}\underline{\hat{\mathbf{h}}}(m) &= \left[\underline{\hat{\mathbf{h}}}_0^T(m) \cdots \underline{\hat{\mathbf{h}}}_{K-1}^T(m)\right]^T \\ &= \left[\underline{\hat{h}}_0(m) \cdots \underline{\hat{h}}_{2L-1}(m)\right]^T.\end{aligned} \tag{31}$$

Since SPMMax-MDF aims to obtain fast convergence with low complexity, our approach of achieving SP tap-selection is then to select $1 \leq M_2 \leq 2L$ elements from $|\underline{\chi}_i(m)\underline{\hat{h}}_i(m)|$ for $i = 0, \dots, 2L-1$, where elements $\underline{\chi}_i(m)$ can be obtained from $\underline{\mathbf{g}}(m)$ defined in (22). Elements of the $2L \times 2L$ diagonal tap-selection matrix $\mathbf{Q}(m)$ are therefore given by

$$q_i(m) = \begin{cases} 1 & i \in \{\text{indices of the } M_2 \text{ maxima of} \\ & \qquad |\underline{\chi}_i(m)\underline{\hat{h}}_i(m)|\}, \\ 0 & \text{otherwise,} \end{cases} \tag{32}$$

for $i = 0, \dots, 2L-1$. Employing (32), the diagonal matrix $\underline{\tilde{\mathbf{D}}}(m-k)$ in (21) for the SP tap-selection can be described by (24) and (25).

It should be noted that additional simulations performed using selection criteria by sorting $|\underline{\chi}_i^*(m)\underline{\chi}_i(m)\underline{\hat{h}}_i(m)/p_i(m)|$ showed no significant improvement for SPMMax-MDF as it was found that the sparseness effect of $|\underline{\hat{h}}_i(m)|$ dominates the selection process compared to the term $\underline{\chi}_i^*(m)\underline{\chi}_i(m)/p_i(m)$, which results in selecting the same filter coefficients for adaptation as would be selected using (32). In addition, normalization by $p_i(m)$ incurs an extra $2L$ divisions, which is not desirable for our VoIP application. As a final comment, since the number of the "active" coefficients of $\underline{\mathbf{h}}$ reduces with increasing K, we choose M_2 to be

$$M_2 = \frac{(2-a)L}{K} + aL. \tag{33}$$

This enables M_2 to reduce with increasing K hence allowing adaptation to be more concentrated on the "active" region. A good choice of a has been found experimentally to be given by $a = 1$. The proposed SPMMax-MDF algorithm is described in Algorithm 1.

3.3. Computational complexity

Although it is well known, from the computational complexity point of view, that $N = L$ is the optimal choice for the MDF algorithm, it nevertheless is more efficient than time-domain implementations even for $N < L$ [8]. As shown in Algorithm 1, the proposed SPMMax-MDF computes $\underline{\tilde{\mathbf{D}}}(m-k)$ using tap-selection matrix $\mathbf{Q}(m)$, which is defined by (26) and (32) for $\text{mod}(m,T) = 0$ and $\text{mod}(m,T) \neq 0$, respectively. We show in Table 1 the number of multiplications and divisions required for MDF, MMax-MDF, MMax-MDF$_\text{N}$, and SPMMax-MDF to compute the term $\underline{\tilde{\mathbf{D}}}^*(m-k)\mathbf{P}^{-1}(m)\underline{\mathbf{e}}(m)$. We have also included the recently proposed IPMDF algorithm [22] for comparison. It should be noted that for MMax and SP tap-selection in (26) and (32), no additional computational complexity is introduced since $|\underline{\chi}_i(m)|$ and $|\underline{\chi}_i(m)\underline{\hat{h}}_i(m)|$ can be obtained from (18) and (17), respectively. For MMax-MDF$_\text{N}$, however, computing the selected filter coefficients for adaptation using (23) incurs additional number of divisions. The complexity for each algorithm for an example case of $L = 512$, $T = 8$, $M_1 = 0.5 \times 2L$, and $K = 64$ is shown in Table 2. It can be seen that the complexity of the proposed SPMMax-MDF is approximately 50% of that for the MDF. Compared to MMax-MDF, SPMMax-MDF requires only an additional 2% of multiplications and divisions. However, as will be shown in Section 4, the performance of SPMMax-MDF is better than MMax-MDF. Finally, the complexity of SPMMax-MDF is 33% and 25% of that for the IPMDF algorithm in terms of multiplications and divisions, respectively.

Table 1: Complexity of algorithms.

Algorithm	Multiplication	Division
MDF	$2L$	$2L$
IPMDF	$3L$	$4L$
MMax-MDF	M_1	M_1
MMax-MDF$_\text{N}$	M_1	$M_1 + 2L$
SPMMax-MDF	$[M_1 + (T-1)M_2]/T$	$[M_1 + (T-1)M_2]/T$

Table 2: Complexity for the case of $L = 512$, $T = 8$, $M_1 = 0.5 \times 2L$, and $K = 64$.

Algorithm	Multiplication	Division
MDF	1024	1024
IPMDF	1536	2048
MMax-MDF	512	512
MMax-MDF$_\text{N}$	512	1536
SPMMax-MDF	519	519

4. RESULTS AND DISCUSSIONS

We present simulation results to illustrate the performance of the proposed SPMMax-MDF algorithm for NEC using a recorded network impulse response $\mathbf{h}$ with 512 taps [12], as shown in Figure 2. The performance is measured using normalized misalignment defined as

$$\eta = \frac{\|\mathbf{h} - \hat{\mathbf{h}}(n)\|_2^2}{\|\mathbf{h}\|_2^2}. \tag{34}$$

We used a sampling frequency of 8 kHz and white Gaussian noise (WGN) $w(n)$ was added to achieve a signal-to-noise ratio (SNR) of 20 dB. The following parameters for the algorithms are chosen for all simulations [22]: $T = 8$, $\lambda = [1 - 1/(3L)]^N$, $\mathbf{S}(0) = \sigma_x^2/100$, $\delta = 20\sigma_x^2 2N/L$. Step-size control variable β has been adjusted for each algorithm so as to achieve the same steady-state performance.

We first compare the variation in convergence of MMax-MDF$_\text{N}$ and MMax-MDF with M_1 using step-size control variables $\beta = 0.7$ and $\beta = 0.6$ for MMax-MDF$_\text{N}$ and MMax-MDF, respectively. We used a CGN input generated

$$
\begin{aligned}
&\delta = 20\sigma_x^2 N/L,\\
&\lambda = \left[1 - \frac{1}{3L}\right]^N,\\
&\mu = \beta(1-\lambda),\ 0 < \beta \leq 1,\\
&\mathbf{S}(0) = \sigma_x^2/100,\\
&\widehat{\mathbf{h}}_k(m) = \left[\widehat{h}_{kN}(m)\widehat{h}_{kN+1}(m)\cdots\widehat{h}_{kN+N-1}(m)\right]^T,\\
&\underline{\widehat{\mathbf{h}}}_k(m) = \mathbf{F}_{2N}\begin{bmatrix}\widehat{\mathbf{h}}_k(m)\\ \mathbf{0}_{N\times 1}\end{bmatrix},\\
&\underline{\mathbf{g}}(m) = \left[\underline{\chi}_0(m)\underline{\chi}_1(m)\cdots\underline{\chi}_{2L-1}(m)\right],\\
&i = 0, 1, \dots, 2L-1,\\
&\textit{MMax tap-selection}\ \text{for}\ \operatorname{mod}(m,T) = 0,\\
&q_i(m) = \begin{cases}1 & i \in \{\text{indices of the } M_1 \text{ maxima of } |\underline{\chi}_i(m)|\},\\ 0 & \text{otherwise},\end{cases}\\
&\textit{SP tap-selection}\ \text{for}\ \operatorname{mod}(m,T) \neq 0,\\
&M_2 = (2-a)L/K + aL,\\
&q_i(m) = \begin{cases}1 & i \in \{\text{indices of the } M_2 \text{ maxima of } |\underline{\chi}_i(m)\underline{\widehat{h}}_i(m)|\},\\ 0 & \text{otherwise},\end{cases}\\
&\underline{\widetilde{\mathbf{g}}}(m) = \mathbf{Q}(m)\underline{\mathbf{g}}(m) = \left[\underline{\widetilde{\chi}}_0(m)\cdots\underline{\widetilde{\chi}}_{2L-1}(m)\right]^T,\\
&\underline{\widetilde{\mathbf{D}}}(m-k) = \operatorname{diag}\{\underline{\widetilde{\chi}}_{2kN}(m)\cdots\underline{\widetilde{\chi}}_{2kN+2N-1}(m)\},\\
&\underline{\mathbf{e}}(m) = \underline{\mathbf{y}}(m) - \mathbf{G}^{01}\textstyle\sum_{k=0}^{K-1}\underline{\mathbf{D}}(m-k)\underline{\widehat{\mathbf{h}}}_k(m-1),\\
&\mathbf{S}(m) = \lambda\mathbf{S}(m-1) + (1-\lambda)\underline{\mathbf{D}}^*(m)\underline{\mathbf{D}}(m),\\
&\mathbf{P}(m) = \mathbf{S}(m) + \delta\mathbf{I}_{2N\times 2N},\\
&\widehat{\mathbf{h}}_k(m) = \underline{\widehat{\mathbf{h}}}_k(m-1) + \mu\mathbf{G}^{10}\underline{\widetilde{\mathbf{D}}}^*(m-k)\mathbf{P}^{-1}(m)\underline{\mathbf{e}}(m).
\end{aligned}
$$

Algorithm 1: The SPMMax-MDF algorithm.

by filtering zero-mean WGN through a lowpass filter with a single pole [12]. It can be seen from Figure 5 that for each case of M_1, the degradation in convergence performance due to tap-selection is less for the MMax-MDF$_{\mathrm{N}}$ than the MMax-MDF. However, as shown in Tables 1 and 2, MMax-MDF$_{\mathrm{N}}$ incurs $2L$ additional divisions compared to the MMax-MDF algorithm.

We next compare the convergence performance of SPMMax-MDF with MDF and IPMDF using CGN input for $K = 1$ in Figure 6. We have used $T = 8$ and $\beta = 0.6$ for all algorithms. We have also used $M_1 = 0.5 \times 2L$ since it was shown in [28] that by such setting, a good balance between complexity reduction and performance degradation due to MMax tap-selection can be reached. As can be seen from the figure, the performance of SPMMax-MDF is close to that for the MDF since for $K = 1$ which results in $M_2 = 2L$ according to (33). Consequently, under the condition of $\operatorname{mod}(m,T) \neq 0$, all the $2L$ filter coefficients are updated, while under the condition of $\operatorname{mod}(m,T) = 0$, $M_1 = 0.5 \times 2L$ coefficients are updated. As a result of this, and consistent with any partial update algorithms presented in [28], the performance of SPMMax-MDF approaches that for the MDF. Compared to IPMDF, SPMMax-MDF only requires approximately 63% and 47% of the number of multiplications and division, as indicated in Table 1.

We show in Figure 7 the convergence performance of SPMMax-MDF, MDF, and IPMDF for $K > 1$ using CGN input. As before, we have used the same step-size control variable of $\beta = 0.6$ for all algorithms except for the cases of SPMMax-MDF, where $\beta = 0.8$ is used to archive the same steady-state performance. It can be seen that for $K = 64$, the proposed SPMMax-MDF algorithm achieves faster rate of convergence in terms of normalized misalignment compared to the more complex MDF during adaptation. Since, as shown in Figure 4, ξ increases with K, it can therefore be expected that such improvement can be increased when larger K is employed. In addition, as the delay for MDF is reduced by a factor of K compared to FLMS, the proposed SPMMax-MDF can archive further delay reduction for larger K and thus is desirable for NEC. For the case of $M_1 = 0.5 \times 2L$ and $K = 64$, the number of multiplications and divisions required for each algorithm is shown in Table 2.

Figure 8 shows the performance of the algorithms obtained using a male speech input. Parameters used for

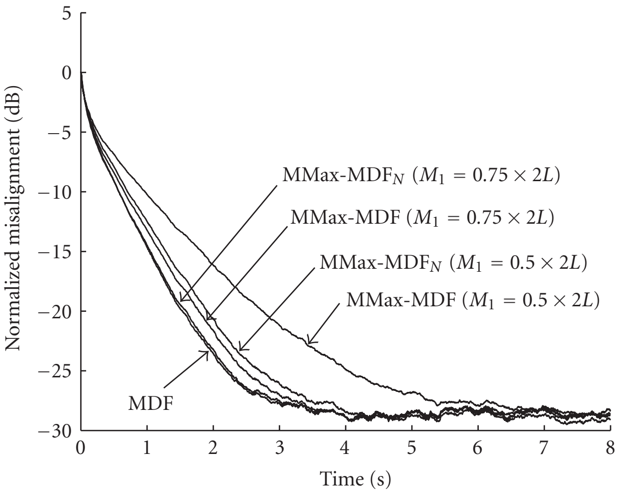


FIGURE 5: Variation of performance with M_1 for MMax-MDF$_N$ and MMax-MDF.

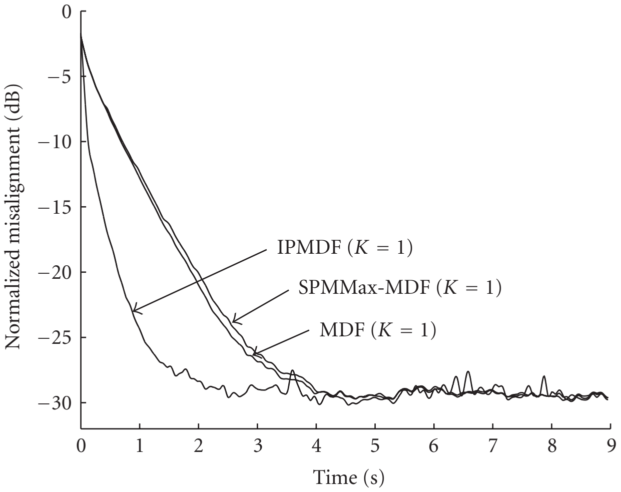


FIGURE 6: Performance of SPMMax-MDF using CGN input for $T = 8$, $M_1 = 0.5 \times 2L$, $K = 1$.

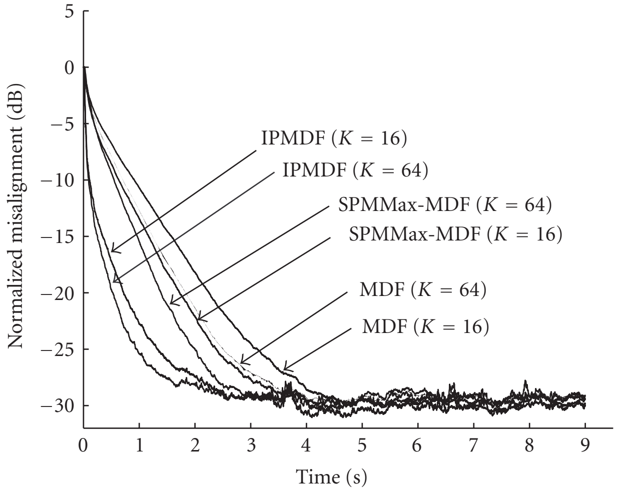


FIGURE 7: Performance of SPMMax-MDF for CGN input with $T = 8$ and $M_1 = 0.5 \times 2L$.

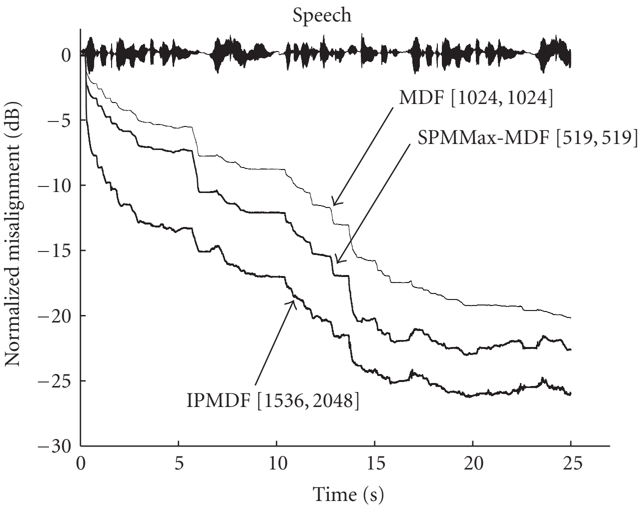


FIGURE 8: Performance of SPMMax-MDF using speech input for $T = 8$, $M_1 = 0.5 \times 2L$, $K = 64$, and the computational complexity required for each algorithm.

each algorithm are the same as that for the previous simulations except that for SPMMax-MDF, where we have used $\beta = 1$ to achieve the same steady-state performance. The computational complexity required for each algorithm is also shown in the figure between square brackets, where the first and the second integers represent the number of multiplications and divisions, respectively. It can be seen that SPMMax-MDF achieves approximately 5 dB improvement in terms of normalized misalignment with lower complexity in comparison to MDF. In addition, the performance of our low cost SPMMax-MDF algorithm approaches that of IPMDF.

5. CONCLUSIONS

We have proposed SPMMax-MDF for network echo cancellation in VoIP. This algorithm achieves a faster rate of convergence, low complexity, and low delay by novelly exploiting both the MMax and SP tap-selection in the frequency domain using MDF implementation. We discussed two approaches of incorporating MMax tap-selection into MDF and showed their tradeoff between rate of convergence and complexity. Simulation results using both colored Gaussian noise and speech inputs show that the proposed SPMMax-MDF achieves up to 5 dB improvement in convergence performance with significantly lower complexity compared to MDF. In addition, the performance of our low cost SPMMax-MDF algorithm approaches that of IPMDF. Since the MDF structure has been applied for acoustic echo cancellation (AEC) [21] and blind acoustic channel identification [29], where the impulse responses are nonsparse, the proposed SPMMax-MDF algorithm can also be potentially applied to these applications for reducing computational complexity and algorithmic delay.

REFERENCES

[1] B. Goode, "Voice over internet protocol (VoIP)," *Proceedings of the IEEE*, vol. 90, no. 9, pp. 1495–1517, 2002.

[2] H. M. Chong and H. S. Matthews, "Comparative analysis of traditional telephone and voice-over-internet protocol (VoIP) systems," in *Proceedings of the IEEE International Symposium on Electronics and the Environment (ISEE '04)*, pp. 106–111, Phoenix, Ariz, USA, May 2004.

[3] H.-G. Kang, H. K. Kim, and R. V. Cox, "Improving the transcoding capability of speech coders," *IEEE Transactions on Multimedia*, vol. 5, no. 1, pp. 24–33, 2003.

[4] G. L. Choudhury and R. G. Cole, "Design and analysis of optimal adaptive de-jitter buffers," *Computer Communications*, vol. 27, no. 6, pp. 529–537, 2004.

[5] A. Raake, "Short- and long-term packet loss behavior: towards speech quality prediction for arbitrary loss distributions," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 6, pp. 1957–1968, 2006.

[6] M. M. Sondhi and D. A. Berkley, "Silencing echoes on the telephone network," *Proceedings of the IEEE*, vol. 68, no. 8, pp. 948–963, 1980.

[7] J. Radecki, Z. Zilic, and K. Radecka, "Echo cancellation in IP networks," in *Proceedings of the 45th International Midwest Symposium on Circuits and Systems (MWSCAS '02)*, vol. 2, pp. 219–222, Tulsa, Okla, USA, August 2002.

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

[9] D. L. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancelers," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 5, pp. 508–518, 2000.

[10] J. Benesty and S. L. Gay, "An improved PNLMS algorithm," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '02)*, vol. 2, pp. 1881–1884, Orlando, Fla, USA, May 2002.

[11] J. Cui, P. A. Naylor, and D. T. Brown, "An improved IPNLMS algorithm for echo cancellation in packet-switched networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '04)*, vol. 4, pp. 141–144, Montreal, Quebec, Canada, May 2004.

[12] H. Deng and M. Doroslovački, "New sparse adaptive algorithms using partial update," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '04)*, vol. 2, pp. 845–848, Montreal, Quebec, Canada, May 2004.

[13] T. Aboulnasr and K. Mayyas, "Complexity reduction of the NLMS algorithm via selective coefficient update," *IEEE Transactions on Signal Processing*, vol. 47, no. 5, pp. 1421–1424, 1999.

[14] S. M. Kuo and D. R. Morgan, "Active noise control: a tutorial review," *Proceedings of the IEEE*, vol. 87, no. 6, pp. 943–973, 1999.

[15] A. Carini and G. L. Sicuranza, "Analysis of transient and steady-state behavior of a multichannel filtered-*x* partial-error affine projection algorithm," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2007, Article ID 31314, 15 pages, 2007.

[16] H. Deng and M. Doroslovački, "Proportionate adaptive algorithms for network echo cancellation," *IEEE Transactions on Signal Processing*, vol. 54, no. 5, pp. 1794–1803, 2006.

[17] E. R. Ferrara, "Fast implementations of LMS adaptive filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 474–475, 1980.

[18] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, 1965.

[19] S. Haykin, *Adaptive Filter Theory*, Information and System Science, Prentice-Hall, Upper Saddle River, NJ, USA, 4th edition, 2002.

[20] J. J. Shynk, "Frequency-domain and multirate adaptive filtering," *IEEE Signal Processing Magazine*, vol. 9, no. 1, pp. 14–37, 1992.

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

[22] A. W. H. Khong, P. A. Naylor, and J. Benesty, "A low delay and fast converging improved proportionate algorithm for sparse system identification," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2007, Article ID 84376, 8 pages, 2007.

[23] I. Pitas, "Fast algorithms for running ordering and max/min calculation," *IEEE Transactions on Circuits and Systems*, vol. 36, no. 6, pp. 795–804, 1989.

[24] P. A. Naylor and W. Sherliker, "A short-sort M-Max NLMS partial-update adaptive filter with applications to echo cancellation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '03)*, vol. 5, pp. 373–376, Hong Kong, April 2003.

[25] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.

[26] J. Benesty, Y. A. Huang, J. Chen, and P. A. Naylor, "Adaptive algorithms for the identification of sparse impulse responses," in *Selected Methods for Acoustic Echo and Noise Control*, E. Hänsler and G. Schmidt, Eds., chapter 5, pp. 125–153, Springer, Berlin, Germany, 2006.

[27] A. W. H. Khong and P. A. Naylor, "Efficient use of sparse adaptive filters," in *Proceedings of the 40th Asilomar Conference on Signals, Systems and Computers (ACSSC '06)*, pp. 1375–1379, Pacific Grove, Calif, USA, October-November 2006.

[28] A. W. H. Khong and P. A. Naylor, "Selective-tap adaptive filtering with performance analysis for identification of time-varying systems," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 5, pp. 1681–1695, 2007.

[29] R. Ahmad, A. W. H. Khong, and P. A. Naylor, "Proportionate frequency domain adaptive algorithms for blind channel identification," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '06)*, vol. 5, pp. V29–V32, Toulouse, France, May 2006.