

# FREQUENCY DOMAIN SELECTIVE TAP ADAPTIVE ALGORITHMS FOR SPARSE SYSTEM IDENTIFICATION

Andy W.H. Khong<sup>1</sup>, Xiang (Shawn) Lin<sup>1</sup>, Miloš Doroslovački<sup>2</sup>, Patrick A. Naylor<sup>1</sup>

<sup>1</sup>Department of Electrical and Electronic Engineering, Imperial College London, UK

<sup>2</sup>Department of Electrical and Computer Engineering, The George Washington University, USA

## ABSTRACT

We propose a new low complexity and fast converging frequency-domain adaptive algorithm for sparse system identification. This is achieved by exploiting the MMax and SP tap-selection criteria for complexity reduction and fast convergence respectively. We incorporate these tap-selection techniques into the multi-delay filtering (MDF) algorithm in order to reduce the delay inherent in frequency-domain algorithms. We illustrate two such approaches and discuss the tradeoff between convergence performance and computational complexity for these approaches. Simulation results show an improvement in convergence rate for the proposed algorithm over MDF with reduced complexity. The proposed algorithm achieves a convergence performance close to that of the recently proposed but substantially more complex improved proportionate MDF algorithm.

**Index Terms**— sparse system identification, network echo cancellation, partial update, frequency-domain adaptive filtering

## 1. INTRODUCTION

Sparse system identification has found applications in packet-switched network echo cancellation (NEC) [1] and geophysical exploration as well as communication systems. In NEC for example, the echo path impulse response is caused by the impedance mismatch between the four- and two-wire circuits connected via the network hybrid. The echo path impulse response is typically of length 64-128 ms and exhibits an “active” region, defined as the region of the impulse response containing large magnitude coefficients, in the range of 8-12 ms duration. The impulse response is dominated by regions where magnitudes are close to zero making the impulse response sparse. These “inactive” regions are due to the presence of bulk delay caused by network propagation, encoding, and jitter buffer delays. For effective echo cancellation, algorithms adopted by network echo cancellers aim to estimate the sparse impulse response. Adaptive algorithms, such as the normalized least-mean-square (NLMS), have been employed successfully for estimating impulse responses in acoustic and communication channels. For the identification of sparse systems, such as occur in NEC as shown in Fig. 1, several algorithms have been proposed which utilize the sparse nature of the impulse response to achieve higher rate of convergence than the NLMS algorithm. One of the first algorithms proposed for NEC is the proportionate NLMS (PNLMS) algorithm [2] where, at each iteration, the step-size for each filter coefficient is proportional to its magnitude. Although PNLMS achieves fast initial convergence, its rate of convergence is reduced significantly by the adaptation of small coefficients [3]. The improved PNLMS (IPNLMS) algorithm [4] addresses this problem by incorporating both proportionate and NLMS adaptation. The

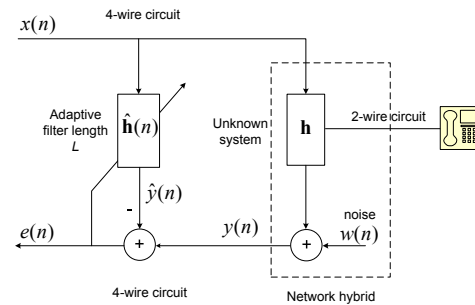


Fig. 1. Network echo cancellation.

IPNLMS algorithm however requires twice as many multiplications per iteration as the NLMS algorithm.

As applications utilizing voice over IP (VoIP) have gained more popularity in recent years, algorithms have been proposed with the aim of achieving complexity reduction. The algorithm proposed in [5] alternates between updating the whole adaptive filter using NLMS and only those coefficients which are significantly large. More recently, the Sparse Partial (SP) update NLMS (SPNLMS) algorithm [6] incorporates the MMax tap-selection [7] for the reduction of computational complexity. In order to achieve an improvement in convergence rate, SPNLMS updates filter coefficients corresponding to both tap-inputs and filter coefficients having large magnitudes. It has been well known that frequency-domain adaptive filtering such as the fast-LMS (FLMS) algorithm [8] offers an attractive means of achieving efficient implementation. One of the main drawbacks of FLMS, however, is the delay introduced between the input and output, which is equivalent to the length of the adaptive filter  $L$ . This delay can be significant since the number of filter coefficients can be several hundreds. To address this problem of delay, the multi-delay filter (MDF) structure has been proposed [9] which partitions the adaptive filter into blocks of length  $N$  independent of  $L$ . The benefit of low delay for MDF over FLMS and NLMS in the context of NEC has been shown in [10].

In this work, we propose a *low complexity* and *fast converging* adaptive algorithm for NEC. We achieve this by incorporating the SP update for fast convergence and the MMax tap-selection for low complexity. In addition, we exploit the MDF structure to achieve *low delay* between the input and output. We first describe how MMax tap-selection can be incorporated into the MDF algorithm using two approaches and we illustrate the tradeoff between their rate of convergence and complexity in Section 3.1. We then propose to incorporate the SP tap-selection into the MMax-MDF algorithm giving (SPMMax-MDF) as described in Section 3.2. We illustrate, in Section 4, the performance of the proposed MMax-MDF and SPMMax-MDF algorithms.

## 2. REVIEW OF SPNLMS AND MDF ALGORITHMS

The Sparse Partial (SP) update NLMS (SPNLMS) algorithm [6] has been proposed to achieve fast convergence with low computational complexity. With reference to Fig. 1, we first define  $\mathbf{x}(n) = [x(n) \dots x(n-L+1)]^T$ ,  $\mathbf{h} = [h_1 \dots h_L]^T$  and  $\hat{\mathbf{h}}(n) = [\hat{h}_1(n) \dots \hat{h}_L(n)]^T$  where  $[\cdot]^T$  is defined as vector transposition. The SPNLMS update equation can be described by

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu \frac{\mathbf{Q}(n)\mathbf{x}(n)e(n)}{\|\mathbf{Q}(n)\mathbf{x}(n)\|^2 + \delta}, \quad (1)$$

where  $e(n) = y(n) - \mathbf{x}^T(n)\hat{\mathbf{h}}(n-1)$ ,  $\|\cdot\|^2$  is the squared  $l_2$ -norm and  $\delta$  is the regularization parameter. The  $L \times L$  tap-selection matrix  $\mathbf{Q}(n) = \text{diag}\{q_1(n) \dots q_L(n)\}$  contains elements  $q_i(n)$  for tap-selection index  $i = 1, \dots, L$ . The SPNLMS utilizes MMax tap-selection [7] to achieve complexity reduction. For fast convergence, coefficients are updated based on the magnitudes of the input and the estimated filter coefficients. The relative significance of these tap-selection strategies is controlled by the variable  $T \in \mathbb{Z}^+$ . Elements of  $\mathbf{Q}(n)$  are then given, under the condition of  $\text{mod}(n, T) = 0$ , by

$$q_i(n) = \begin{cases} 1 & i \in \{M_1 \text{ maxima of } |x(n-i+1)|\}, \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

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

$$q_i(n) = \begin{cases} 1 & i \in \{M_2 \text{ maxima of } |x(n-i+1)\hat{h}_i(n)|\}, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

The variables  $M_1$  and  $M_2$  define the number of selected taps for MMax and the active taps selected for sparse adaptation respectively. The SPNLMS incorporates MMax tap-selection given by (2) for complexity reduction. To achieve fast convergence for sparse systems, SPNLMS selects filter coefficients corresponding to the largest products of tap-input and filter coefficients given by (3).

The MDF algorithm [9] was proposed to address the problem of delay inherent in FLMS [8] since the latter computes the output for every  $L$  samples. The MDF addresses this problem by partitioning the adaptive filter of length  $L$  into  $K$  subfilters each of length  $N$  with  $L = KN$ . Consequently, the delay of MDF is reduced by a factor of  $L/N$  compared to FLMS. To describe the MDF algorithm, we first define  $m$  as the frame index and the following time-domain quantities given by  $\mathbf{X}(m) = [\mathbf{x}(mN) \dots \mathbf{x}(mN+N-1)]$ ,  $\mathbf{y}(m) = [y(mN) \dots y(mN+N-1)]^T$ ,  $\hat{\mathbf{h}}(m) = [\hat{\mathbf{h}}_0^T(m) \dots \hat{\mathbf{h}}_{K-1}^T(m)]^T$ ,  $\hat{\mathbf{y}}(m) = [\hat{y}(mN) \dots \hat{y}(mN+N-1)]^T = \mathbf{X}^T(m)\hat{\mathbf{h}}(m)$ ,  $\mathbf{e}(m) = \mathbf{y}(m) - \hat{\mathbf{y}}(m)$  and the  $2N \times 1$  input vector

$$\boldsymbol{\chi}(m-k) = [x(mN-kN-N) \dots x(mN-kN+N-1)]^T, \quad (4)$$

where  $k = 0, \dots, K-1$  is defined as the block index and the subfilters  $\hat{\mathbf{h}}_k(m) = [\hat{h}_{kN+1}(m) \dots \hat{h}_{kN+N}(m)]^T$ . We next define  $\mathbf{F}$  as the Fourier matrix and a  $2N \times 2N$  diagonal matrix

$$\underline{\mathbf{D}}(m-k) = \text{diag}\{\mathbf{F}\boldsymbol{\chi}(m-k)\} = \text{diag}\{\underline{\boldsymbol{\chi}}(m-k)\} \quad (5)$$

with elements containing the Fourier transform of  $\boldsymbol{\chi}(m-k)$  for the  $k^{\text{th}}$  subfilter. We also define the following frequency-domain quantities  $\underline{\mathbf{y}}(m) = \mathbf{F} \begin{bmatrix} \mathbf{0}_{N \times 1} \\ \mathbf{y}(m) \end{bmatrix}$ ,  $\underline{\hat{\mathbf{h}}}_k(m) = \mathbf{F} \begin{bmatrix} \hat{\mathbf{h}}_k(m) \\ \mathbf{0}_{N \times 1} \end{bmatrix}$ ,  $\underline{\mathbf{e}}(m) = \mathbf{F} \begin{bmatrix} \mathbf{0}_{N \times 1} \\ \mathbf{e}(m) \end{bmatrix}$ ,  $\mathbf{G}^{01} = \mathbf{F}\mathbf{W}^{01}\mathbf{F}^{-1}$ ,  $\mathbf{W}^{01} = \begin{bmatrix} \mathbf{0}_{N \times N} & \mathbf{0}_{N \times N} \\ \mathbf{0}_{N \times N} & \mathbf{I}_{N \times N} \end{bmatrix}$ ,

$\mathbf{G}^{10} = \mathbf{F}\mathbf{W}^{10}\mathbf{F}^{-1}$  and  $\mathbf{W}^{10} = \begin{bmatrix} \mathbf{I}_{N \times N} & \mathbf{0}_{N \times N} \\ \mathbf{0}_{N \times N} & \mathbf{0}_{N \times N} \end{bmatrix}$  where  $\mathbf{0}_{N \times N}$

is an  $N \times 1$  null matrix. The MDF algorithm is then given by [9]

$$\underline{\mathbf{e}}(m) = \underline{\mathbf{y}}(m) - \mathbf{G}^{01} \sum_{k=0}^{K-1} \underline{\mathbf{D}}(m-k)\hat{\underline{\mathbf{h}}}_k(m-1), \quad (6)$$

$$\mathbf{S}_{\text{MDF}}(m) = \lambda \mathbf{S}_{\text{MDF}}(m-1) + (1-\lambda)\underline{\mathbf{D}}^*(m)\underline{\mathbf{D}}(m), \quad (7)$$

$$\hat{\underline{\mathbf{h}}}_k(m) = \hat{\underline{\mathbf{h}}}_k(m-1) + \mu \mathbf{G}^{10} \underline{\mathbf{D}}^*(m-k) \times [\mathbf{S}_{\text{MDF}}(m) + \delta_{\text{MDF}}]^{-1} \underline{\mathbf{e}}(m), \quad (8)$$

where  $*$  denotes complex conjugate,  $0 \ll \lambda < 1$  is the forgetting factor and  $\mu = \beta(1-\lambda)$  is the step-size with  $0 < \beta \leq 1$ . Letting  $\sigma_x^2$  be the input signal variance, the initial regularization parameters [11] are  $\mathbf{S}_{\text{MDF}}(0) = \sigma_x^2/100$  and  $\delta_{\text{MDF}} = 20\sigma_x^2N/L$ . For  $N = L$ ,  $K = 1$  and hence MDF is equivalent to FLMS [8].

## 3. THE SPARSE PARTIAL UPDATE MULTI-DELAY FILTERING ALGORITHM

We propose to incorporate the MMax and SP tap-selection into the MDF algorithm. Similar to SPNLMS, we aim to achieve reduced complexity due to MMax tap-selection as well as improved convergence performance due to SP tap-selection. In addition, we exploit the low delay provided by the MDF structure. We first describe how the MMax tap-selection in (2) can be incorporated into the MDF which we denote MMax-MDF. We next show the SPMMax-MDF algorithm in which SP tap-selection given by (3) is incorporated into the MMax-MDF. The proposed MMax-MDF and SPMMax-MDF algorithms can be described by (6), (7) and

$$\hat{\underline{\mathbf{h}}}_k(m) = \hat{\underline{\mathbf{h}}}_k(m-1) + \mu \mathbf{G}^{10} \tilde{\underline{\mathbf{D}}}^*(m-k) \times [\mathbf{S}_{\text{MDF}}(m) + \delta_{\text{MDF}}]^{-1} \underline{\mathbf{e}}(m). \quad (9)$$

The difference between (8) and (9) 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.

### 3.1. The MMax-MDF Algorithms

The diagonal matrix  $\tilde{\underline{\mathbf{D}}}(m-k)$  for MMax-MDF can be obtained using two approaches. When tap-selection is performed in the time-domain, we denote the algorithm as MMax-MDF<sub>t</sub>. To indicate tap-selection in the frequency-domain, we use MMax-MDF<sub>f</sub>. For the time-domain selection approach, elements for the  $2N \times 2N$  diagonal tap-selection matrix  $\mathbf{Q}(m)$  can then be expressed by subselecting from elements in  $\boldsymbol{\chi}(m)$  defined in (4) giving, for  $1 \leq M_1 \leq 2N$ ,

$$q_i(m) = \begin{cases} 1 & i \in \{M_1 \text{ maxima of } |x(j-N+i-1)|\}, \\ 0 & \text{otherwise,} \end{cases} \quad (10)$$

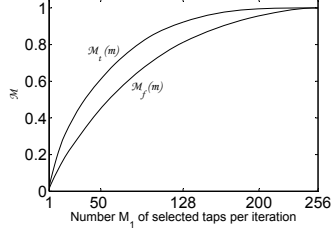
with  $j = mN - kN$  and  $i = 1, \dots, 2N$ . Utilizing  $\mathbf{Q}(m)$ , elements of  $\tilde{\underline{\mathbf{D}}}(m-k)$  for the MMax-MDF<sub>t</sub> algorithm can be expressed as

$$\tilde{\underline{\mathbf{D}}}(m-k) = \text{diag}\{\mathbf{F}\mathbf{Q}(m-k)\boldsymbol{\chi}(m-k)\}. \quad (11)$$

The MMax-MDF<sub>f</sub> algorithm is described by (6), (7), (11) and (9).

For frequency-domain selection, the approach is to select frequency bins corresponding to the largest magnitude Fourier transform of the tap-input over all the subfilter blocks  $k = 0, \dots, K-1$ . Defining  $\underline{\boldsymbol{\chi}}(m-k) = \mathbf{F}\boldsymbol{\chi}(m-k)$ , the concatenated Fourier transform of the input across all subfilters is then defined as

$$\underline{\mathbf{g}}(m) = [\underline{\boldsymbol{\chi}}^T(m) \dots \underline{\boldsymbol{\chi}}^T(m-K+1)]^T = [\underline{\chi}_1(m) \dots \underline{\chi}_{2L}(m)]^T. \quad (12)$$



**Fig. 2.** Variation of  $\mathcal{M}_t(m)$  and  $\mathcal{M}_f(m)$  with  $M_1$  for  $K = 1$  and  $L = N = 128$ .

Each element in the tap-input matrix  $\mathbf{Q}(m)$  is then given by

$$q_i(m) = \begin{cases} 1 & i \in \{M_1 \text{ maxima of } |\chi_i(m)|\}, \\ 0 & \text{otherwise,} \end{cases} \quad (13)$$

for  $i = 1, \dots, 2L$  and for this case of frequency-domain selection  $1 \leq M_1 \leq 2L$ . Defining a  $2L \times 1$  vector  $\tilde{\mathbf{g}}(m)$  containing the subselected Fourier transform of the input vector as

$$\tilde{\mathbf{g}}(m) = \mathbf{Q}(m)\mathbf{g}(m) = [\tilde{\chi}_1(m) \dots \tilde{\chi}_{2L}(m)]^T, \quad (14)$$

the  $2N \times 2N$  diagonal matrix  $\tilde{\mathbf{D}}(m-k)$  for this frequency-domain tap-selection approach is then given by

$$\tilde{\mathbf{D}}(m-k) = \text{diag}\{\tilde{\chi}_{2kN+1}(m) \dots \tilde{\chi}_{2kN+2N}(m)\}. \quad (15)$$

The MMax-MDF<sub>f</sub> algorithm is described by (6), (7), (15) and (9).

We next compare the convergence performance of MMax-MDF<sub>t</sub> and MMax-MDF<sub>f</sub>. It has been shown [12], that the convergence performance of MMax-NLMS degrades with reducing normalized energy of the subselected tap-input vector brought about by  $M_1$ . This energy is defined by  $\mathcal{M}(n) = \|\mathbf{Q}(n)\mathbf{x}(n)\|^2 / \|\mathbf{x}(n)\|^2$  with elements of  $\mathbf{Q}(n)$  given in (2). In the same manner and for the case of  $N = L$ , convergence performance of MMax-MDF<sub>t</sub> and MMax-MDF<sub>f</sub> can be studied by defining respectively for the time- and frequency-domain tap-selection approaches,

$$\mathcal{M}_t(m) = \|\mathbf{Q}(m)\chi(m)\|^2 / \|\chi(m)\|^2, \quad (16)$$

$$\mathcal{M}_f(m) = \|\mathbf{F}^{-1}\mathbf{Q}(m)\tilde{\mathbf{g}}(m)\|^2 / \|\mathbf{F}^{-1}\tilde{\mathbf{g}}(m)\|^2, \quad (17)$$

where  $\mathbf{Q}(m)$  in (16) and (17) are defined by (10) and (13) respectively. Due to the orthogonality property of the Fourier transform, matrix  $\mathbf{F}^{-1}$  can be omitted from (17). Figure 2 shows how  $\mathcal{M}_t(m)$  and  $\mathcal{M}_f(m)$  vary with  $M_1$  for the case of  $N = L = 128$  with a white Gaussian noise (WGN) input  $\chi(m)$ . Since  $\mathcal{M}_t(m) > \mathcal{M}_f(m)$  when  $M_1 < 2L$ , we would then expect that the degradation in performance due to tap-selection is less for time-domain selection in MMax-MDF<sub>t</sub> than frequency-domain in MMax-MDF<sub>f</sub>. We further illustrate this by way of simulation presented in Section 4.

Although selection in the time-domain induces a less significant degradation in convergence performance than for the frequency-domain, the computational cost for the latter is lower. This is because the diagonal elements in  $\tilde{\mathbf{D}}(m-k)$  given by (15) for MMax-MDF<sub>f</sub> consists of  $2L - M_1$  null elements across all  $k$ . On the other hand, due to matrix  $\mathbf{F}$ , diagonal elements in  $\mathbf{D}(m-k)$  given by (11) for MMax-MDF<sub>t</sub> does not necessarily contain null elements.

### 3.2. The Sparse Partial Update MMax-MDF Algorithm

The proposed SPMMMax-MDF algorithm utilizes the SP tap-selection defined by (3) to achieve fast convergence for the identification of

**Table 1.** Complexity of algorithms

Algorithm	Multiplication	Division
MDF	$2L$	$2L$
IPMDF	$3L$	$4L$
SPMMMax-MDF	$[M_1 + (T-1)M_2]/T$	$[M_1 + (T-1)M_2]/T$

**Table 2.** Complexity for the case of  $L = 512$ ,  $T = 8$ ,  $M_1 = 0.0625 \times 2L$  and  $M_2 = L$

Algorithm	Multiplication	Division
MDF	1024	1024
IPMDF	1536	2048
SPMMMax-MDF	456	456

sparse impulse responses. Incorporating the SP tap-selection into MDF is not trivial, since as can be seen from (4), the length of the input frame  $\chi(m)$  is  $2N$  compared to  $L$  for the adaptive filter. To address this problem, we first define a  $2L \times 1$  vector obtained by concatenating all subfilters  $\hat{\mathbf{h}}_k(m)$  for  $k = 0, \dots, K-1$ , i.e.,

$$\hat{\mathbf{h}}(m) = [\hat{\mathbf{h}}_0^T(m) \dots \hat{\mathbf{h}}_{K-1}^T(m)]^T = [\hat{h}_1(m) \dots \hat{h}_{2L}(m)]^T. \quad (18)$$

Our approach of achieving SP tap-selection is then to select  $1 \leq M_2 \leq 2L$  elements from  $|\chi_i(m)\hat{h}_i(m)|$  for  $i = 1, \dots, 2L$ , where elements  $\chi_i(m)$  can be obtained from  $\mathbf{g}(m)$  defined in (12). This selection technique is then equivalent to subselecting frequency bins corresponding to the  $M_2$  magnitude response where both  $\chi_i(m)$  and  $\hat{h}_i(m)$  are significantly large. Elements of the  $2L \times 2L$  diagonal tap-selection matrix  $\mathbf{Q}(m)$  are then given by

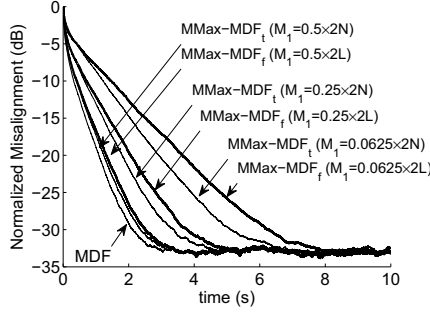
$$q_i(m) = \begin{cases} 1 & i \in \{M_2 \text{ maxima of } |\chi_i(m)\hat{h}_i(m)|\}, \\ 0 & \text{otherwise,} \end{cases} \quad (19)$$

for  $i = 1, \dots, 2L$ . The diagonal matrix  $\tilde{\mathbf{D}}(m-k)$  in (9) for the SP tap-selection can be described by (14) and (15).

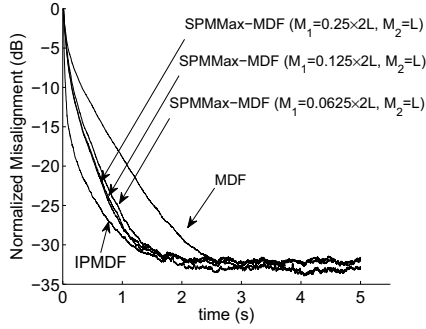
The proposed SPMMMax-MDF utilizes both MMax and SP tap-selection as described above. Since SPMMMax-MDF aims to achieve fast convergence with low complexity, we utilize MMax-MDF<sub>f</sub> when  $\text{mod}(m, T) = 0$ . We show in Section 4 that any degradation due to MMax tap-selection can be offset by SP tap-selection. The proposed SPMMMax-MDF is described by (6), (7) and (9) where  $\tilde{\mathbf{D}}(m-k)$  is defined as (15). For  $\text{mod}(m, T) = 0$  and  $\text{mod}(m, T) \neq 0$ ,  $\mathbf{Q}(m)$  is defined by (13) and (19) respectively. The number of multiplications and divisions required for MDF [9], IPMDF [10] and SPMMMax-MDF to compute the term  $\tilde{\mathbf{D}}^*(m-k)[\mathbf{S}_{\text{MDF}}(m) + \delta_{\text{MDF}}]^{-1}\mathbf{e}(m)$  are as shown in Table 1. For an example case of  $L = 512$ ,  $T = 8$ ,  $M_1 = 0.0625 \times 2L$  and  $M_2 = L$ , the complexity is as shown in Table 2.

## 4. SIMULATION RESULTS

We compare the performance of the proposed algorithms for NEC using a recorded network impulse response  $\mathbf{h}$  similar to that as shown [6]. The adaptive filter  $\hat{\mathbf{h}}(m)$  is chosen to be of the same length as  $\mathbf{h}$  with  $L = 512$  and we define the normalized misalignment as  $\eta(m) = \|\mathbf{h} - \hat{\mathbf{h}}(m)\|^2 / \|\mathbf{h}\|^2$ . We used a sampling frequency of 8 kHz and additive WGN  $w(n)$  is added to achieve a signal-to-noise ratio (SNR) of 20 dB. The following parameters for the algorithms are chosen for all simulations [10]:  $N = 8$ ,



**Fig. 3.** Variation of performance with  $M_1$  for MMax-MDF<sub>f</sub> and MMax-MDF<sub>t</sub>.

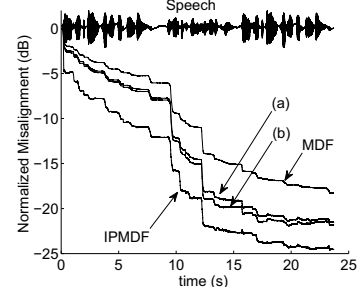


**Fig. 4.** Performance of SPMMMax-MDF for WGN input with  $T = 8$ .

$K = 64$ ,  $T = 8$ ,  $\lambda = [1 - 1/(3L)]^N$ ,  $\mathbf{S}_{\text{MDF}}(0) = \sigma_x^2/100$ ,  $\delta_{\text{MDF}} = \sigma_x^2 20N/L$ . We first illustrate how the convergence of MMax-MDF<sub>t</sub> and MMax-MDF<sub>f</sub> vary with  $M_1$  using WGN input and a step-size control variable  $\beta = 0.6$ . It can be seen from Fig. 3 that for each case of  $M_1$ , the degradation in convergence of MMax-MDF<sub>t</sub> due to tap-selection is lower than that for MMax-MDF<sub>f</sub>. This is because, as explained in Section 3.1,  $\mathcal{M}_t(m) > \mathcal{M}_f(m)$ . For both algorithms, since  $\mathcal{M}_t(m)$  and  $\mathcal{M}_f(m)$  reduces with  $M_1$ , the rate of convergence reduces with  $M_1$  as expected.

Figure 4 shows the convergence performance of the proposed SPMMMax-MDF algorithm where, as explained in Section 3.2, we perform selection in the frequency-domain in order to reduce computational complexity. We have used  $T = 8$  and  $\beta = 1$  for SPMMMax-MDF. For each iteration,  $M_2 = L$  frequency-bins are selected for SP adaptation. For the MDF algorithm, we have used  $\beta = 0.6$  in order to achieve the same steady-state performance as SPMMMax-MDF. For purpose of comparison, we have also included the convergence of the substantially more complex IPMDF algorithm [10]. It can be seen that the proposed SPMMMax-MDF algorithm achieves higher rate of convergence of approximately 6 dB in terms of normalized misalignment compared to the more complex MDF during adaptation. For the case of  $M_1 = 0.0625 \times 2L$  and  $M_2 = L$ , the number of multiplications required for each algorithm is shown in Table 2.

Figure 5 shows the performance of the algorithms obtained using a male speech input. Parameters used for each algorithms are the same as that for the WGN input. The performance of SPMMMax-MDF with  $M_1 = 0.0625 \times 2L$  and  $M_2 = L$  is shown in Fig. 5 (a) while Fig. 5 (b) shows the case when  $M_1 = 0.25 \times 2L$  and  $M_2 = L$ . We note that SPMMMax-MDF achieves approximately 6 dB improvement in normalized misalignment with lower complexity than that for MDF. As before, the convergence of SPMMMax-MDF is close to the substantially more complex IPMDF algorithm.



**Fig. 5.** Performance of SPMMMax-MDF using speech input for  $T = 8$  with (a)  $M_1 = 0.125L$ ,  $M_2 = L$  and (b)  $M_1 = 0.5L$ ,  $M_2 = L$ .

## 5. CONCLUSION

We have proposed SPMMMax-MDF for sparse system identification. This algorithm achieves high rate of convergence with low complexity by novelly exploiting both the MMax and SP tap-selection. 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 WGN and speech inputs show that the proposed SPMMMax-MDF achieves approximately 6 dB improvement in convergence performance with only 45% of the complexity of MDF. The performance of SPMMMax-MDF is close to that for the substantially more complex IPMDF algorithm.

## 6. REFERENCES

- [1] S. L. Gay and J. Benesty, *Acoustic Signal Processing for Telecommunication*. Kluwer Academic Publishers, 2001.
- [2] D. L. Duttweiler, "Proportionate normalized least mean square adaptation in echo cancellers," *IEEE Trans. Speech Audio Processing*, vol. 8, no. 5, pp. 508–518, Sep. 2000.
- [3] A. Deshpande and S. L. Grant, "A new multi-algorithm approach to sparse system adaptation," in *Proc. European Signal Process. Conf.*, 2005.
- [4] J. Benesty and S. L. Gay, "An improved PNLMS algorithm," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 2, 2002, pp. 1881–1884.
- [5] T. Gänslar, J. Benesty, M. M. Sondhi, and S. L. Gay, "Dynamic resource allocation for network echo cancellation," in *Int. Conf. Acoust. Speech Signal Processing*, vol. 5, May 2001, pp. 3233–3236.
- [6] H. Deng and M. Doroslovacki, "New sparse adaptive algorithms using partial update," in *Proc. IEEE Int. Conf. on Signal Processing*, vol. 2, May 2004, pp. 845–848.
- [7] T. Aboulnasr and K. Mayyas, "Complexity reduction of the NLMS algorithm via selective coefficient update," *IEEE Trans. Signal Processing*, vol. 47, no. 5, pp. 1421–1424, 1999.
- [8] E. R. Ferrara, "Fast implementations of LMS adaptive filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 28, pp. 474–475, 1980.
- [9] J. S. Soo and K. K. Pang, "Multidelay block frequency domain adaptive filter," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, no. 2, pp. 373–376, Feb. 1990.
- [10] A. W. H. Khong, J. Benesty, and P. A. Naylor, "A low delay and fast converging improved proportionate algorithm for sparse system identification," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2007, 2007.
- [11] J. Benesty, T. Gänslar, D. R. Morgan, M. M. Sondhi, and S. L. Gay, *Advances in Network and Acoustic Echo Cancellation*. Springer, 2001.
- [12] A. W. H. Khong and P. A. Naylor, "Selective-tap adaptive filtering with performance analysis for identification of time-varying systems," *IEEE Trans. Audio Speech Language Processing*, vol. 15, no. 5, pp. 1681–1695, Jul. 2007.