In this paper, we propose a novel frequency domain adaptive tap partial update algorithm for network echo cancellation. Controlled by a simple sparseness measurement of the filter coefficients, the new algorithm adapts not only the filter coefficients, but also the number of taps to be updated iteration by iteration. Using the convergence behavior of the sparseness measurement, the proposed algorithm regards the convergence of the filter coefficients as a two-stage process incorporating the MMax and SPMMax partial update criteria together with the improved proportionate multi-delay filtering (IPMDF) and multi-delay filtering (MDF) adaptive algorithms. Simulation results show that, compared with fully updated IPMDF, our proposed algorithm achieves the same convergence performance with significantly reduced computational complexity.

Index Terms—network echo cancellation, sparse system, frequency domain adaptive filtering, partial update, multi-delay filtering, improved proportionate multi-delay filtering

1. INTRODUCTION

Research on network echo cancellation (NEC) [1] has been an active area for several decades. The network echo results from the impedance mismatch of the hybrid connected between a four-wire central network circuit and a two-wire local network circuit. In NEC, a typical duration of the echo path impulse response is 64 to 128 ms, and the active region, which is defined as the region where the impulse response coefficients have large magnitude, is relatively small. Figure 1(B) shows a network echo path impulse response sampled at 8 kHz. To cancel the echo or to identify the impulse response of the echo path, the Least Mean Square (LMS) and Normalized LMS (NLMS) techniques may be employed as the adaptive algorithms shown in Figure 1(A). However, for the case of a long and sparse channel, those algorithms have the disadvantages of slow convergence speed and high power consumption. To improve the convergence speed, Proportionate NLMS (PNLMS) [2] and improved PNLMS (IPNLMS) [3] have been proposed. By introducing a step-size control matrix, PNLMS significantly increases the convergence rate compared with NLMS. However, PNLMS is not applicable for a non-sparse impulse response. This problem is addressed in IPNLMS using a controllable combination of both the NLMS and PNLMS algorithms. On the other hand, partial update algorithms are proposed to reduce the computational complexity. By taking advantage of the sparse nature of the echo path impulse response, partial update algorithms, such as MMax [4] and Sparse Partial

Figure 1. Network echo canceller schematic diagram (A) and a sparse echo impulse response (B).

MMax (SPMMax) selective tap adaptive algorithms [5], adapt only a small subset of the filter taps in each iteration. With an appropriate tap selection criterion, they can achieve similar or even better convergence performance with reduced complexity.

Recently frequency domain adaptive algorithms have gained attention because of their efficient implementations and delay performance improvements. Previously, fast-LMS (FLMS) [6] was considered as an attractive alternative to the time-domain algorithms. This method implements an efficient block-LMS updating strategy through the use of the FFT. However, the FLMS has the drawback of introducing a long delay between the input and the output. To address this delay problem, a flexible MDF structure was proposed [7], where the L-tap adaptive filter is partitioned into an arbitrary number K of sub-filters having a smaller block size N. To further improve the convergence speed, [8] incorporates a proportionality control of the IPNLMS algorithm together with the frequency domain MDF structure, and proposes the improved proportionate MDF (IPMDF). It can be observed from simulation results shown in [8] that IPMDF achieves both an improvement in the convergence performance as well as a better time-varying tracking ability compared with MDF and IPNLMS. The primary limitation of IPMDF is its computational complexity.

Previously, a partial update MDF structure, SPMMax-MDF was designed in [9] by applying the SPMMax partial update criterion to MDF. However, without a priori knowledge of the network echo path, it is difficult to determine the optimal subset of taps to update in order to guarantee convergence. Inspired by the convergence nature of a sparseness quantity, we proposed a time-domain adaptive tap partial update filter for IPNLMS in [10] which can adaptively change the number of filter taps being updated according to a real-time sparseness value. In this paper, our objective is to design a low complexity, fast converging and efficient frequency-domain partial update algorithm using the
adaptive tap strategy of [10]. We also divide the convergence process of the estimated filter coefficients into two stages, before and after the convergence of the sparseness measure, using the MMax-IPMDF and SPMax-MDF structures, respectively. Simulation results show that our proposed frequency domain adaptive algorithm achieves the same convergence performance as fully updated IPMDF while only updating a small number of taps, which reduces both computational complexity and power consumption.

2. REVIEW OF FREQUENCY DOMAIN ADAPTIVE ALGORITHMS AND PARTIAL UPDATE CRITERIA

2.1. MDF and IPMDF

MDF was proposed in order to address the inherent delay problem of previous frequency domain adaptive algorithms [7]. This partitions the adaptive filter of length \( N \) into \( K \) smaller delay sub-filters having block size \( N \) so that \( L = KN \). It reduces the delay of FLMS by a factor of \( K \). We first define \( m \) as the frame index, \( k \) as the sub-filter index, and the following time domain quantities:

\[
x(mN) = [x(mN), \ldots, x(mN-L+1)]^T,
\]

\[
X(m) = [x(mN), \ldots, x(mN+N-1)]^T,
\]

\[
y(m) = [y(mN), \ldots, y(mN+N-1)]^T,
\]

\[
h(m) = [h_0(T(m)), \ldots, h_{K-1}(T(m))]^T, \quad \hat{h}(m) = \hat{h}_N(m), \ldots, \hat{h}_{N+K-1}(m)^T, \quad \hat{y}(m) = [\hat{y}(mN), \ldots, \hat{y}(mN+N-1)]^T = X^T(m)\hat{h}(m),
\]

\[
e(m) = y(m) - \hat{y}(m) = [e(mN), \ldots, e(mN+N-1)]^T.
\]

To describe the MDF algorithm, let \( F \) and \( F^{-1} \) be the \( 2N \times 2N \) Fourier Matrix and inverse Fourier Matrix. The \( 2N \times 1 \) input vector to the \( k^{th} \) sub-filter is

\[
D(m-k) = \text{diag}[D_2(m), \ldots, D_2(m)] = \text{diag}[\hat{y}(m), \ldots, \hat{y}(m)] = \text{diag}[\hat{y}(m), \ldots, \hat{y}(m)] = \text{diag}[\hat{y}(m), \ldots, \hat{y}(m)],
\]

where \( \hat{y}_k(m) = [x(mN-kN-N), \ldots, x(mN-kN+N-1)]^T \) is the \( m^{th} \) input vector overlapped with the \( (m-1)^{th} \) input vector. We also define the following frequency domain quantities:

\[
y(m) = F[y(m)], \quad \hat{h}_k(m) = F[h_k(m)], \quad e(m) = F[e(m)],
\]

\[
G_{01} = FW_0F^{-1}, \quad G_{10} = FW_1F^{-1}.
\]

Then the MDF frequency domain adaptive algorithm is as in [8]:

\[
\hat{e}(m) = \hat{y}(m) - G_{01} \sum_{k=0}^{K-1} D(m-k)\hat{h}_k(m-1), \quad S(m) = \lambda S(m-1) + (1-\lambda)D*(m)D(m),
\]

\[
\hat{h}_k(m) = \hat{h}_k(m-1) + \mu G_{10} D*(m-k)S(m) + \delta^2 e(m), \quad 0 < \lambda \leq 1 \text{ is the forgetting factor and } \mu = \beta(1-\lambda) \text{ is the step size with } 0 < \beta \leq 1 \text{ [8].} \]

As an enhancement of MDF, IPMDF [8] incorporates the idea of proportionate step-size control of IPNLMS [3]. It introduces a time-domain step-size control matrix \( Q_{k}(m) \) to each sub-filter:

\[
Q_{k}(m) = \text{diag}[q_{k}(m)], \quad q_{k}(m) = [q_{KN}(m), \ldots, q_{KN+N-1}].
\]

\[
q_{KN+j}(m) = \frac{1 - \alpha}{2L} + (1 + \alpha) \frac{\|\hat{e}(m)\|}{\|h_j\| + \varepsilon}, \quad \varepsilon > 0
\]

with a small number \( \varepsilon \) used to prevent dividing by zero, and \( \alpha \in [-1, 1] \) is an adjustable parameter similar to that used in IPNLMS. The update scheme for IPMDF is now generalized as:

\[
\hat{h}_k(m) = \hat{h}_k(m-1) + \mu G_{1}Q_{k}(m)G_{2}D*(m-k)[S(m) + \delta^2 e(m)],
\]

where \( G_{1} = F[I_{N \times N}, 0_{N \times N}] \) and \( G_{2} = [I_{N \times N}, 0_{N \times N}]F^{-1} \). To achieve the same steady-state error, the regularization parameters \( S(0) \) in (4) and \( \delta \) in (5) need to be scaled by \( 1 - \alpha \) for IPMDF. Simulations [8] show significant convergence improvement of IPMDF over MDF.

2.2. MMax and SPMax Partial Update Criteria

Partial update adaptive filters have been of great interest due to their ability to save power consumption. MMax is widely used because of its simplicity, while achieving comparable performance compared with fully updated adaptive algorithms. For partial update NLMS, the updating scheme is described as:

\[
\hat{h}(m) = \hat{h}(m-1) + \mu \left[ \frac{P(m)x(m)e(m)}{\|P(m)x(m)\|_2^2 + \delta} \right],
\]

where \( P(m) = \text{diag}[p_0(m), \ldots, p_{L-1}(m)] \) determines the portion of taps being updated. For the MMax scheme:

\[
p(m) = \begin{cases} 1, & \text{if } m \text{ corresponds to } M_1 \text{ maxima} \\ 0, & \text{else} \end{cases}
\]

MMax updates the filter coefficients corresponding to the \( M_1 \) largest input values. To achieve faster convergence, the SPMax criterion incorporates the idea of updating filter coefficients corresponding to the \( M_2 \) largest products of the input and the filter, as shown below:

\[
p(m) = \begin{cases} 1, & \text{if } m \text{ corresponds to } M_2 \text{ maxima of } \left[ x(m-l+1)h_l(m) \right] \\ 0, & \text{else} \end{cases}
\]

if \( \text{mod}(m, T) = 0 \): \( p(m) \) updates as (9).

The quantities \( M_1, M_2, T \) are arbitrary numbers; how to select their optimum values remains an open question for all partial update algorithms. This motivates us to design a technique for an adaptive tap partial update filter that is controlled by the sparseness of the estimated filter coefficients.

3. PROPOSED SPARSENESS-CONTROLLED ADAPTIVE TAP PARTIAL UPDATE IPMDF
3.1. Definition of Sparseness

The sparseness of a vector \( \mathbf{h} \) of length \( L \) may be defined as [11]:

\[
\xi = \frac{L}{L - \sqrt{L}} \left( 1 - \frac{\| \mathbf{h} \|_2}{\sqrt{L}} \right).
\] (11)

It can be observed that \( 0 \leq \xi \leq 1 \) for any given \( \mathbf{h} \), with equality satisfied for a delta channel \( (\xi = 1) \) and a constant channel \( (\xi = 0) \) respectively regardless of the length \( L \). In addition, for sparse network echo cancellation, as the estimated filter coefficients \( \hat{\mathbf{h}}(m) \) gradually converge to their optimal values, the sparseness measure \( \hat{\xi}(m) \) of the estimated filter coefficients will also converge to its optimal value, which is the sparseness of the echo path impulse response. Indeed, \( \hat{\xi}(m) \) is expected to converge even faster than \( \hat{\mathbf{h}}(m) \) does, as it is less sensitive to the fluctuation of \( \hat{\mathbf{h}}(m) \) around its optimal value due to the stochastic gradient characteristic. Here, we use a modified version of the sparseness measure \( \hat{s}(m) \) as in [10]:

\[
\hat{s}(m) = \left( 1 - \frac{\| \hat{\mathbf{h}}(m) \|_2}{\sqrt{L}} \right)^P + \zeta.
\] (12)

Here, the pre-factor in (11) has been removed and an exponent of \( P \) is introduced to allow trade-offs between performance and power consumption. \( \zeta \) is a small value used to prevent dividing by zero in the initial iteration; therefore, \( 0 < \hat{s}(m) < 1 \).

3.2. Two-Stage Adaptive Tap Strategy

We use the sparseness measure \( \hat{s}(m) \) of the time domain estimated filter coefficients to adaptively control the number of taps that need updating in the frequency domain in each iteration:

\[
M'(m) = 2L \times (1 - \hat{s}(m)).
\] (13)

Note that the total number of corresponding frequency domain taps is \( 2L \) for an \( L \)-tap time domain adaptive filter. It has been shown in [10] that, with \( P \) set to 1, \( M'(m) \) converges quickly to a small number for the delta channel, while for the constant channel our adaptive strategy reduces to a full tap algorithm. The sparseness-controlled adaptive tap strategy works well over a wide range of impulse responses. The larger the sparseness measure value, the smaller the number of partial update taps.

The convergence process of the estimated coefficients can be divided into two stages, i.e. before and after the convergence of the sparseness measure. If \( |\hat{s}(m) - \hat{s}(m - 1)| \geq \gamma \), then \( \hat{s}(m) \) has not yet converged, and extra attention is needed for the coefficients having large magnitude to ensure a fast initial convergence speed. Otherwise, \( \hat{s}(m) \) can be viewed as having converged. Here \( \gamma \) is a small value used to differentiate the two stages and we have found that a value of 0.0001 gives good results.

3.3. Proposed Partial Update Algorithm

Based on the ideas that have been described in Sections 3.1 and 3.2, we propose a new frequency domain adaptive tap partial update algorithm. The convergence process is regarded as a combination of two stages, where different frequency domain adaptive algorithms and different partial update criteria are used. First let us define the \( 2N \times 1 \) vectors \( \hat{\mathbf{h}}(m) \) and \( \chi(m) \) in the frequency domain:

\[
\hat{\mathbf{h}}(m) = [\hat{h}_0^T(m), ..., \hat{h}_{K-1}^T(m)]^T = [\hat{h}_0(m), ..., \hat{z}_{2L-1}(m)]^T,
\]

\[
\chi(m) = [\chi_0^T(m), ..., \chi_{K-1}^T(m)]^T = [\chi_0(m), ..., \chi_{2L-1}(m)]^T.
\]

Our adaptive tap partial update IPMDF algorithm is described as:

1) 1st Stage \( : |\hat{s}(m) - \hat{s}(m - 1)| \geq \gamma \)

\[
M'(m) = 2L \times (1 - \hat{s}(m))
\]

\[
p_1(m) = \begin{cases} 1, & \text{if } l \text{ corresponds to } M'(m) \text{ maxima} \\ 0, & \text{else} \end{cases}
\] (14)

\[
P_\chi(m) = \text{diag} \left( P_{\chi N}(m), ..., P_{\chi N+N-1}(m) \right)
\] (15)

\[
\tilde{\mathbf{D}}(m-k) = P_\chi(m) \mathbf{D}(m-k)
\] (16)

\[
\hat{\mathbf{h}}_{\chi}(m) = \hat{\mathbf{h}}_{\chi}(m-1) + L_\mu \mathbf{G}_1 \chi(m) \mathbf{G}_2 \hat{\mathbf{D}}^*(m-k) \times \left[ \mathbf{S}(m) + \delta \right]^{-1} \mathbf{e}(m)
\] (17)

2) 2nd Stage \( : |\hat{s}(m) - \hat{s}(m - 1)| < \gamma \)

\[
M'(m) \text{ remain the same as last } M'(m) \text{ of the 1st stage}
\]

if mod(m,T) \( \neq 0 \) : \( p_1(m) \) updates as (14)

if mod(m,T) \( = 0 \) :

\[
p_1(m) = \begin{cases} 1, & \text{if } l \text{ corresponds to } M'(m) \text{ maxima} \\ 0, & \text{else} \end{cases}
\] (18)

\[
P_\chi(m) = \text{diag} \left( P_{\chi N}(m), ..., P_{\chi N+N-1}(m) \right)
\] (19)

\[
\tilde{\mathbf{D}}(m-k) = P_\chi(m) \mathbf{D}(m-k)
\] (20)

\[
\hat{\mathbf{h}}_{\chi}(m) = \hat{\mathbf{h}}_{\chi}(m-1) + L_\mu \mathbf{G}_1 \hat{\mathbf{D}}^*(m-k) \mathbf{S}(m) + \delta \mathbf{e}(m)
\] (21)

In the 1st stage, the MMMax-IPMDF algorithm is used for faster initial convergence with filter coefficients considered in the sorting process (\( M'(m) \) maxima of \( \chi(m) \)) instead of \( \chi(m) \). In the 2nd stage, when the sparseness measure reaches a steady state value, all filter coefficients fluctuate around their optimal values. Then, the SPMMMax-MDF algorithm [9] is sufficient to achieve good performance. Since \( \hat{s}(m) \) rarely changes in the 2nd stage, calculation of \( \hat{s}(m) \) is unnecessary and the number of taps updating in the 2nd stage remains the same as the last \( M'(m) \) of the 1st stage. Note that the diagonal matrix \( \mathbf{D}(m-k) \) in (17) and (21) contain \( 2L - M' \) null elements across all \( k \), which significantly reduces the computational complexity compared with full-tap IPMDF.

4. SIMULATION RESULTS

The performance is evaluated using the normalized misalignment:
The average number of taps that are updating for our algorithm is 552 (about $0.5 \times 2L$) out of 1024 (2L). However, it still achieves the same convergence speed as fully updated IPMDF (and is thus faster than SPMMAX-MDF). Moreover, the trade-off between convergence performance and computational complexity may be adjusted by choosing different values for $P$.

5. CONCLUSIONS

In this paper, we have proposed a novel frequency-domain partial update algorithm which adapts both the number of taps which are updated as well as the filter coefficients. The convergence process is implemented as a two-stage process using the frequency-domain MM-MAX-IPMDF and SPMMAX-MDF procedures, respectively, together with our adaptive tap strategy. The simulation results show that, compared with fully updated IPMDF, our algorithm achieves the same convergence speed with a significantly reduced computational complexity.

6. REFERENCES