IJICT International Journal of Information & Communication Technology
ITRC

Volume 2- Number 2- July 2010 (pp.61-70)

Research Note (CT Section)

A Unified Approach to Set-Membership and Selective Partial Update Adaptive Filtering Algorithms

Mohammad Shams Esfand Abadi
Faculty of Electrical and Computer Engineering
Shahid Rajaee Teacher Training University
Tehran, Iran
mshams@srttu.edu

Hamid Palangi
Department of Electrical Engineering
Sharif University of Technology
Tehran, Iran
hmd.palangi@gmail.com

Received: February 28, 2009- Accepted: July 20, 2010

Abstract—**In this paper we show how the classical and modern adaptive filter algorithms can be introduced in a unified way. The Max normalized least mean squares (MAX-NLMS), *N*-Max NLMS, the family of SPU-NLMS, SPU transform domain adaptive filter (SPU-TDAF), and SPU subband adaptive filter (SPU-SAF) are particular algorithms are established in a unified way. Following this, the concept of set-membership (SM) adaptive filtering is extended to this framework, and a unified approach to derivation of SM and SM-SPU adaptive filters is presented. The SM-NLMS, SM-TDAF, SM-SAF, SM-SPU-NLMS, and SM-SPUSAF are presented based on this approach. Also, this concept is extended to the SPU affine projection (SPU-AP) and SPUTDAF algorithms and two new algorithms which are called SM-SPU-AP and SM-SPU-TDAF algorithms, are established. These novel algorithms are computationally efficient. The good performance of the presented algorithms is demonstrated in system identification application.**

Keywords- Adaptive filter, selective partial update, set-membership.

I. Introduction

Adaptive filtering is an important subfield of digital signal processing having numerous applications [1], [2], [3], [4]. In some of these applications, a large number of filter coefficients are needed to achieve an acceptable performance. Therefore the computational complexity is one of the main problems in these applications. Several adaptive filter algorithms such as the subband adaptive filters (SAFs), the adaptive filter algorithms with selective partial updates (SPU) and the setmembership (SM) filtering have been proposed to solve this problem.

The SPU adaptive algorithms update only a subset of the filter coefficients in each time iteration and consequently reduce the computational complexity. The Max normalized least mean squares (Max-NLMS) [5], the *N*-Max NLMS [6], [7], variants of the selective partial update normalized least mean square (SPU-NLMS) [8], [9], [10], [11], the SPU transform domain LMS (SPU-TDAF) [12], the SPU affine projection algorithm (SPU-APA), [13], [14], and selective partial update subband adaptive filter (SPU-SAF) [15] are important examples of this family of adaptive filter algorithms. Unfortunately, as with many other adaptive filter algorithms, the step-size determines the tradeoff between steady-state mean square error (MSE) and convergence rate in SPU adaptive filters.

Having fast convergence, low steady-state MSE, and low computational complexity at the same time is highly desirable. The set-membership normalized LMS (SM-NLMS) is one of the algorithms that has these three features [16]. Based on [16], different SM adaptive algorithms have been developed. The SM affine projection algorithm (SM-APA) [17], [18], and the SM binormalized data-reusing LMS (SM-BNDRLMS) algorithms [19] are important examples of this family of adaptive filters. In [20], the SM-PU-NLMS was presented based on the combination of the partial updating and setmembership filtering approaches to achieve a more computationally efficient algorithm with reasonable performance.

In [21], the subband adaptive algorithm called normalized SAF (NSAF) was developed based on a constrained optimization problem. In [15], the concepts of SM filtering and SPU adaptation were also extended to NSAF, and SM-SAF, SPUSAF, and SM-SPU-SAF algorithms were developed which were computationally efficient.

In this paper we present a unified approach to establishment of the classical and modern adaptive filter algorithms. Accordingly, we present a unified approach to derivation of SM, and SM-SPU adaptive filtering algorithms. What we propose in this paper can be summarized as follows:

- The establishment of the classical and SPU adaptive filter algorithms in a unified way. The NLMS, ²-NLMS, the family of APA, TDAF, Max-NLMS, N-Max NLMS, the family of SPU-NLMS, SPU-TDAF, and SPU-SAF are established through this approach.
- The establishment of a unified approach to derivation of the SM adaptive filter algorithms. The SM-NLMS, SM-APA, SM-TDAF, SM-SAF algorithms are established based on this approach.
- A unified combination of the SPU and SM approaches to derivation of existing SM-SPU adaptive filter algorithms. The SM-SPU-NLMS, and SM-SPU-SAF are established with this approach.
- The establishment of two new adaptive algorithms which are called, SM-SPU-APA, and SM-SPU-TDAF.
- Demonstrating of the presented algorithms in system identification application.

We have organized our paper as follows: In the following section we briefly review the NLMS, the SPU-NLMS and the SM-NLMS algorithms. In the next section the generic filter update equations are introduced. In Section IV, the classical and SPU adaptive filter algorithms are derived. In Section V, a unified approach to derivation of SM adaptive filtering is presented. Section VI presents the relations for SM-SPU adaptive filter algorithms based on the unified approach. The computational complexity of the presented algorithms is studied in Section VII. Finally, we present several simulation results to demonstrate the good performance of the SM-SPUAP and SM-SPU-TDAF algorithms. Throughout the paper, the following notations are adopted:

$\lvert.\rvert$	Norm of a scalar.
$\lVert.\rVert$	Squared Euclidean norm of a vector.
$(.)^T$	Transpose of a vector or a matrix.
$Tr(.)$	Trace of a matrix.

$diag(.)$ has the same meaning as the MATLAB operator with the same name: If its argument is a vector, a diagonal matrix with the the diagonal elements given by the vector argument results. If the argument is a matrix, its diagonal is extracted into a resulting vector.

II. Background on NLMS, SPU-NLMS and SM-NLMS algorithms

Figure 1 shows a typical adaptive filter setup, where $x(n)$, $d(n)$ and $e(n)$ are the input, the desired and output error signals, respectively. Here, $\mathbf{h}(n)$ is a $M \times 1$ column vector of filter coefficients at iteration n.

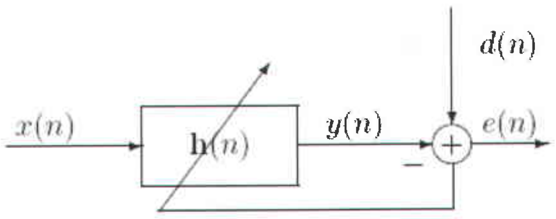

Fig. 1. A typical adaptive filter setup

It is well known that the NLMS algorithm can be derived from the solution of the following optimization problem:

$$\min_{\mathbf{h}(n+1)} \| \mathbf{h}(n+1) - \mathbf{h}(n) \|^2 \tag{1}$$

subject to

$$d(n) = \mathbf{x}^T(n)\mathbf{h}(n+1) \tag{2}$$

where $\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-M+1)]^T$. Using the method of Lagrange multipliers to solve this optimization problem leads to the following recursion:

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \frac{\mu}{\| \mathbf{x}(n) \|^2} \mathbf{x}(n) e(n) \tag{3}$$

where $e(n) = d(n) - \mathbf{x}^T(n)\mathbf{h}(n)$, and μ is the step-size which is introduced in (3) to control over the convergence speed and excess MSE (EMSE).

Now partition the input signal vector and the vector of filter coefficients into K blocks each of length L[1] which are defined as

$$\mathbf{x}(n) = [\mathbf{x}_1^T(n), \mathbf{x}_2^T(n), \ldots, \mathbf{x}_K^T(n)]^T \tag{4}$$

$$\mathbf{h}(n) = [\mathbf{h}_1^T(n), \mathbf{h}_2^T(n), \ldots, \mathbf{h}_K^T(n)]^T \tag{5}$$

The SPU-NLMS algorithm for a single block update in every iteration, minimizes following optimization problem:

$$\min_{\mathbf{h}_j(n+1)} \| \mathbf{h}_j(n+1) - \mathbf{h}_j(n) \|^2 \tag{6}$$

[1] Note that $K = \dfrac{M}{L}$ and is an integer

subject to (2), where j denotes the index of the block that should be updated. Again by using the method of Lagrange multipliers, the update equation for SPU-NLMS is given by:

$$\mathbf{h}_j(n+1)=\mathbf{h}_j(n)+\frac{\mu}{\|\mathbf{x}_j(n)\|^2}\mathbf{x}_j(n)e(n) \tag{7}$$

where $j=\arg\max\|\mathbf{x}_i(n)\|^2$ for $1\le i\le K$ [9].

The SM-NLMS algorithm minimizes (1) subject to $\mathbf{h}(n+1)\in\Psi_n$ where[2]:

$$\Psi_n=\{\mathbf{h}\in\mathbf{R}^M:|d(n)-\mathbf{x}^T(n)\mathbf{h}|\le\gamma\} \tag{8}$$

This aim is achieved by an orthogonal projection of the previous estimate of $\mathbf{h}$ onto the closest boundary of Ψ_n [16]. The resulted recursion for the SM-NLMS is given by:

$$\mathbf{h}(n+1)=\mathbf{h}(n)+\frac{\alpha(n)}{\|\mathbf{x}(n)\|^2}\mathbf{x}(n)e(n) \tag{9}$$

where

$$\alpha(n)=\begin{cases}1-\dfrac{\gamma}{|e(n)|} & if\,|e(n)|>\gamma\\ 0 & Otherwise\end{cases} \tag{10}$$

III. THE GENERIC ADAPTIVE FILTER UPDATE EQUATIONS

From [22], the generic filter vector update equation can be stated as:

$$\mathbf{h}(n+1)=\mathbf{h}(n)+\mu\mathbf{C}(n)\mathbf{X}(n)\mathbf{e}(n) \tag{11}$$

where

$$e(n)=\mathbf{d}(n)-\mathbf{X}^T(n)\mathbf{h}(n) \tag{12}$$

is the output error vector. The matrix $\mathbf{X}(n)$ is a $M\times P$ input signal matrix which is defined as[3]:

$$\mathbf{X}(n)=[\mathbf{x}(n),\mathbf{x}(n-1),\ldots,\mathbf{x}(n-(P-1))] \tag{13}$$

and $\mathbf{d}(n)$ is a $P\times 1$ vector of desired signal which is given by:

$$\mathbf{d}(n)=[d(n),d(n-1),\ldots,d(n-(P-1))]^T \tag{14}$$

The matrix $\mathbf{C}(n)$ is a $M\times M$ invertible matrix called the *preconditioner*. Selecting $\mathbf{C}(n)$ as an approximate inverse of autocorrelation matrix, we can improve the convergence speed dramatically relative to the case when no preconditioner is employed [22]. One strategy for selecting the matrix $\mathbf{C}(n)$ is to use the regularized inverse of the estimated autocorrelation matrix as a preconditioner. In this case, by using the matrix inversion lemma, we write:

$$\mathbf{C}(n)\mathbf{X}(n)=\mathbf{X}(n)\mathbf{W}(n) \tag{15}$$

where $\mathbf{W}(n)$ is the $P\times P$ invertible matrix called the *weighting* matrix [23], [24]. The interested reader is referred to [22] for more details. From this, one might argue that in some cases, a suitable alternative form of the generic adaptive filter is given by:

$$\mathbf{h}(n+1)=\mathbf{h}(n)+\mu\mathbf{X}(n)\mathbf{W}(n)e(n) \tag{16}$$

To have both forms of the generic update equations, we prefer to call the following equation as generic adaptive filter:

$$\mathbf{h}(n+1)=\mathbf{h}(n)+\mu\mathbf{C}(n)\mathbf{X}(n)\mathbf{W}(n)\mathbf{e}(n) \tag{17}$$

IV. DERIVATION OF ADAPTIVE FILTER ALGORITHMS

We can now make specific choices for the preconditioner matrix $\mathbf{C}(n)$ or the weighting matrix $\mathbf{W}(n)$ as well as for the parameter P. Different adaptive filter algorithms can be viewed as specific instantiations of the generic adaptive filter update equation (equations (11), (16), or (17)) [22], [23], [25].

A. *Derivation of the NLMS, ε-NLMS and the Family of Affine Projection Algorithms*

By substituting the matrices $\mathbf{C}(n),\mathbf{W}(n)$, and the parameter P from Table I in the generic adaptive filter update equations, the NLMS, and the ε-NLMS are established respectively. From the generic adaptive filter update equation, we can also derive the family of affine projection algorithms. These algorithms are the standard version of APA, the regularized APA (R-APA) [26], and the binormalized data-reusing LMS (BNDR-LMS) [27].

B. *Derivation of the TDAF and SAF Algorithms*

The transform domain adaptive filter (TDAF) algorithm [2] is also established based on the parameter selections from Table I [22][4]. For the subband adaptive filters $\mathbf{C}(n)=\mathbf{I}$,

$$\mathbf{W}(n)=\mathbf{F}[\varepsilon\mathbf{I}+diag\{diag\{\mathbf{F}^T\mathbf{X}^T(n)\mathbf{X}(n)\mathbf{F}\}\}]^{-1}\mathbf{F}^T$$

and the input signal matrix and the vector of desired signal are defined as

$$\mathbf{X}(n)=[\mathbf{x}(nS),\mathbf{x}(nS-1),\ldots,\mathbf{x}(nS-(D-1))]$$

and

$$\mathbf{d}(n)=[d(nS),d(nS-1),\ldots,d(nS-(D-1)]^T$$

respectively where $\mathbf{F}$ is the $D\times S$ matrix whose columns are the unit responses of the channel filters of the analysis filter bank where, S is the number of subbands and D is the length of the channel filters [22]. It is interesting to note that the adaptive filter algorithms in [28], [29], [21], while derived from

[2] The set Ψ_n is referred to as the constraint set, and its boundaries are hyperplanes. Also, γ is the magnitude of the error bound

[3] The parameter P is a positive integer (usually, but not necessarily $P\le M$)

[4] The matrix $\mathbf{T}$ is an $\mathbf{M}\times\mathbf{M}$ orthogonal transform matrix

different points of view, are the same [24]. Selecting $\varepsilon = 0$ in $\mathbf{W}(n)$, results in (8) from [29].

C. Derivation of the Family of Adaptive Filter Algorithms with Selective Partial Updates

From (17), the generic filter coefficients update equation for $P = 1$, and $\mathbf{W}(n) = \mathbf{I}$, can be stated as:

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \mu \mathbf{C}(n) x(n) e(n) \qquad (18)$$

As we noted, in the adaptive filter algorithms with selective partial updates, the $M \times 1$ vector of filter coefficients is partitioned into K blocks each of length L and in each iteration a subset of these blocks is updated. For this family of adaptive filters, the matrix $\mathbf{C}(n)$ can be obtained from Table II, where the $\mathbf{A}$ matrix is the $M \times M$ diagonal matrix with the $\mathbf{1}$ and $\mathbf{0}$ blocks each of length L on the diagonal and the positions of 1's on the diagonal determine which coefficients should be updated in each iteration. In Table II, the parameter L is the length of the block, K is the number of blocks, and N is the number of blocks to update. Through the specific choices for L, N, and the matrix $\mathbf{C}(n)$, different adaptive filter algorithms with selective partial updates are established.

By partitioning the regressor vector $\mathbf{x}(n)$ into K blocks each of length L, the positions of $\mathbf{1}$ blocks (N blocks and $N \le K$) on the diagonal of $\mathbf{A}$ matrix for each iteration in the family of SPU-NLMS adaptive algorithms are determined by the following procedure:

1) The $\| \mathbf{x}_j(n) \|^2$ values are sorted for $0 \le j \le K-1$, where

$$\mathbf{x}_j(n) = [x(n-jL), x(n-jL-1), \ldots, x(n-jL-(L-1))]$$

2) The j values that determine the positions of $\mathbf{1}$ blocks correspond to the N largest values of $\| \mathbf{x}_j(n) \|^2$

From [22] the filter coefficients update equation for transform domain adaptive filters (TDAF) can be stated as:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{\Lambda}^{-2} \mathbf{u}(n) e(n) \qquad (19)$$

where $\mathbf{w}(n) = \mathbf{T}^T \mathbf{h}(n), \mathbf{u}(n) = \mathbf{T}^T \mathbf{x}(n)$ are the transformed matrix, and $\mathbf{\Lambda}^2$ is the power matrix of transform outputs. This matrix can be estimated by

$$\mathbf{\Lambda}^{-2}(n) = [diag\{diag\{\textstyle\sum_{i=0}^{n} \lambda^{n-i} \mathbf{u}(i) \mathbf{u}^T(i)\}\}]^{-1}$$

where $0 << \lambda < 1$. By partitioning $\mathbf{u}(n)$ into K blocks each of length L and $\mathbf{\Lambda}^{-2}(n)$ into K matrices each of size $L \times L$, the positions of $\mathbf{1}$ blocks (N blocks and $N \le K$) on the diagonal of matrix $\mathbf{A}$ for each iteration in the SPU-TDAF adaptive algorithm are determined by the following procedure [12]:

1) The $\mathbf{u}_j^T(n) \mathbf{\Lambda}_j^{-2}(n) \mathbf{u}_j(n)$ values are sorted for $0 \le j \le K-1$, where $\mathbf{\Lambda}^{-2}(n) = diag\{\mathbf{\Lambda}_0^{-2}(n), \ldots, \mathbf{\Lambda}_{K-1}^{-2}(n)\}$ and

$$\mathbf{u}_j(n) = [u(n-jL), u(n-jL-1), \ldots, u(n-jL-L+1)]^T$$

2) The j values that determine the positions of $\mathbf{1}$ blocks correspond to the N largest values of $\mathbf{u}_j^T(n) \mathbf{\Lambda}_j^{-2}(n) \mathbf{u}_j(n)$.

Therefore, SPU-TDAF adaptive algorithm can be written as:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{A} \mathbf{\Lambda}^{-2}(n) \mathbf{u}(n) e(n) \qquad (20)$$

Multiplying both sides of this equation by T from the left, we obtain:

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \mu \mathbf{T} \mathbf{A} \mathbf{\Lambda}^{-2}(n) \mathbf{T}^T \mathbf{x}(n) e(n) \quad (21)$$

which is in the form of (18) with $\mathbf{C}(n) = \mathbf{T} \mathbf{A} \mathbf{\Lambda}^{-2}(n) \mathbf{T}^T$. The particular choices and their corresponding algorithms are summarized in Tables I and II.

TABLE I
THE MOST COMMON FAMILIES OF ADAPTIVE FILTER ALGORITHMS CAN BE DESCRIBED THROUGH

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \mu \mathbf{C}(n) \mathbf{X}(n) \mathbf{W}(n) \mathbf{e}(n)$$

Algorithm	P	$\mathbf{C}(n)$ *and* $\mathbf{W}(n)$
NLMS	$P = 1$	$\mathbf{C}(n) = [\frac{1}{\|\mathbf{x}(n)\|^2}]\mathbf{I}$ *and* $\mathbf{W}(n) = \mathbf{I}$ *or* $\mathbf{W}(n) = [\frac{1}{\|\mathbf{x}(n)\|^2}]\mathbf{I}$ *and* $\mathbf{C}(n) = \mathbf{I}$
ε -NLMS	$P = 1$	$\mathbf{C}(n) = (\varepsilon\mathbf{I} + \mathbf{x}(n)\mathbf{x}^T(n))^{-1}$ *and* $\mathbf{W}(n) = \mathbf{I}$ *or* $\mathbf{W}(n) = (\varepsilon + \mathbf{x}^T(n)\mathbf{x}(n))^{-1}\mathbf{I}$ *and* $\mathbf{C}(n) = \mathbf{I}$
APA	$P \le M$	$\mathbf{W}(n) = (\mathbf{X}^T(n)\mathbf{X}(n))^{-1}$ *and* $\mathbf{C}(n) = \mathbf{I}$
BNDR-LMS	$P = 2$	$\mathbf{W}(n) = (\mathbf{X}^T(n)\mathbf{X}(n))^{-1}$ *and* $\mathbf{C}(n) = \mathbf{I}$
R-APA	$P \le M$	$\mathbf{C}(n) = (\varepsilon\mathbf{I} + \mathbf{X}(n)\mathbf{X}^T(n))^{-1}$ *and* $\mathbf{W}(n) = \mathbf{I}$ *or* $\mathbf{W}(n) = (\varepsilon\mathbf{I} + \mathbf{X}^T(n)\mathbf{X}(n))^{-1}$ *and* $\mathbf{C}(n) = \mathbf{I}$
TDAF	$P = 1$	$\mathbf{C}(n)=\mathbf{T}[diag\{diag\{\sum_{i=0}^{n}\lambda^{n-i}\mathbf{T}^T\mathbf{x}(i)\mathbf{x}^T(i)\mathbf{T}\}\}]^{-1}\mathbf{T}$ *and* $\mathbf{W}(n)=\mathbf{I}$

TABLE II
FAMILY OF ADAPTIVE FILTERS WITH SELECTIVE PARTIAL UPDATES

Algorithm	P	L	K	N	$\mathbf{C}(n)$
Max-NLMS [5]	1	1	M	1	$\frac{\mathbf{A}}{\|\mathbf{Ax}(n)\|^2}$
N-Max-NLMS [6],[7]	1	1	M	$N\le M$	$\frac{\mathbf{A}}{\|\mathbf{x}(n)\|^2}$
SPU-NLMS[8]	1	L	M /L	$N\le K$	$\frac{\mathbf{A}}{\|\mathbf{x}(n)\|^2}$
SPU-NLMS[9],[30]	1	1	M	$N\le M$	$\frac{\mathbf{A}}{\|\mathbf{Ax}(n)\|^2}$
SPU-NLMS[9]	1	L	M /L	1	$\frac{\mathbf{A}}{\|\mathbf{Ax}(n)\|^2}$
SPU-NLMS[9]	1	L	M /L	$N\le K$	$\frac{\mathbf{A}}{\|\mathbf{Ax}(n)\|^2}$
SPU-TDAF[12]	1	L	M /L	$N\le K$	$\mathbf{TA}\Lambda^{-2}(n)\mathbf{T}^T$

D. *Derivation of the SPU-SAF Algorithms*

From [15], the filter update equation for SPU-SAF can be stated as:

$$\mathbf{h}(n+1)=\mathbf{h}(n)+\mu\mathbf{AX}(n)\mathbf{F}[\mathbf{\Gamma}(n)]^{-1}\mathbf{F}^T\mathbf{e}(n) \quad (22)$$

where

$$\mathbf{\Gamma}(n)=diag\{diag\{\mathbf{F}^T(n)\mathbf{X}^T(n)\mathbf{A}^T\mathbf{AX}(n)\mathbf{F}\}\}$$

and $\mathbf{e}(n)=\mathbf{d}(n)-\mathbf{X}^T(n)\mathbf{h}(n)$.

The $\mathbf{A}$ matrix is again the $M\times M$ diagonal matrix with the $\mathbf{1}$ and $\mathbf{0}$ blocks each of length L on the diagonal and the positions of 1's on the diagonal determine which coefficients should be updated in each subband at every adaptation. In [15], we proposed two criteria to find these positions. The positions of the $\mathbf{1}$ blocks are determined by the following procedure:

1) Partition $\mathbf{X}(n)\mathbf{F}$ into K matrices each of size $L\times S$ as:

$$\mathbf{X}(n)\mathbf{F}=\begin{pmatrix}\mathbf{X}_0(n)\\ \mathbf{X}_1(n)\\ \cdot\\ \cdot\\ \mathbf{X}_{K-1}(n)\end{pmatrix} \quad (23)$$

2) Compute the following values:

$$Tr(\mathbf{X}_j^T(n)\mathbf{X}_j(n)) \;\; for\; 0\le j\le K-1 \quad (24)$$

3) The positions of $\mathbf{1}$ blocks on the diagonal correspond to the N largest values of (24).

Another strategy to find the positions of $\mathbf{1}$ blocks can be summarized as follows:

1) Compute the following values:

$$\mathbf{e}^T(n)\mathbf{F}[diag\{diag\{\mathbf{X}_j^T(n)\mathbf{X}_j(n)\}\}]^{-1}\mathbf{F}^T\mathbf{e}(n) \quad for\; 0\le j\le K-1 \quad (25)$$

2) The positions of $\mathbf{1}$ blocks on the diagonal correspond to the N smallest values of (25).

Comparing (22) with the generic filter vector update equation, we find that with substituting $\mathbf{C}(n)=\mathbf{AI}$, and $\mathbf{W}(n)=\mathbf{F}[\mathbf{\Gamma}(n)]^{-1}\mathbf{F}^T$, the SPU-SAF is established.

V. SET-MEMBERSHIP ADAPTIVE FILTER ALGORITHMS

A. *Set-Membership Affine Projection (SM-AP) Algorithms*

From [17], we know that the SM-APA minimizes (1) subject to $\mathbf{h}\in\mathbf{\Psi}_n\cap\mathbf{\Psi}_{n-1}\cap\ldots\cap\mathbf{\Psi}_{n-P+1}$ where:

$$\mathbf{\Psi}_{n-i}=\{\mathbf{h}\in\mathbf{R}^M:|d(n-i)-\mathbf{x}^T(n-i)\mathbf{h}|\le\gamma\} \quad (26)$$

In [17], it has been shown that the suitable update equation for SM-APA can be stated as:

$$\mathbf{h}(n+1)=\mathbf{h}(n)+\mathbf{X}(n)(\mathbf{X}^T(n)\mathbf{X}(n))^{-1}\mathbf{q}\alpha(n)\mathbf{e}(n) \quad (27)$$

where $\mathbf{q}=[1,0,\ldots,0]^T$ and $\alpha(n)$ can be obtained from (10). It is important to note that the SM-NLMS in [16], and SMBNDR-LMS in [19] can also be established when $P=1$, and $P=2$ respectively.

B. *Set-Membership TDAF (SM-TDAF) Algorithms*

By defining $\mathbf{z}(n)=\mathbf{\Lambda w}(n)$, the SM-TDAF minimizes the following objective function:

$$\min_{\mathbf{z}(n+1)}\|\mathbf{z}(n+1)-\mathbf{z}(n)\|^2 \quad (28)$$

subject to $\mathbf{z}(n)\in\mathbf{\Phi}_n$ where:

$$\mathbf{\Phi}_n=\{\mathbf{z}\in\mathbf{R}^M:|d(n)-g^T(n)\mathbf{z}|\le\gamma\} \quad (29)$$

and $\mathbf{g}(n)=\mathbf{\Lambda}^{-1}\mathbf{u}(n)$. This aim is achieved by an orthogonal projection of the previous estimate of $\mathbf{z}$ onto the closest boundary of $\mathbf{\Phi}_n$. Doing this, the recursion for the SM-TDAF is given by:

$$\mathbf{z}(n+1)=\mathbf{z}(n)+\frac{\alpha(n)}{\|\mathbf{g}(n)\|^2}\mathbf{g}(n)e(n) \quad (30)$$

where $\alpha(n)$ is given by (10). Because the entries of $\mathbf{g}(n)$ are approximately uncorrelated, thanks to the transform $\mathbf{T}$, the normalization in the update term of (30) does not provide any improvement in the convergence rate and can be dropped safely. After dropping the normalization and left multiplying by Λ^{-1} we obtain:

$$\mathbf{w}(n+1)=\mathbf{w}(n)+\alpha(n)\mathbf{\Lambda}^{-2}\mathbf{u}(n)e(n) \quad (31)$$

By multiplying both sides of this equation by $\mathbf{T}$ from the left, we obtain:

$$\mathbf{h}(n+1)=\mathbf{h}(n)+\alpha(n)\mathbf{C}(n)\mathbf{x}(n)e(n) \quad (32)$$

where $\mathbf{C}(n)$ corresponds to the TDAF algorithm and is obtained from Table I.

C. *Set-Membership SAF (SM-SAF) Algorithm*

By defining $\mathbf{X}(n)\mathbf{F}=[\mathbf{x}_0(n),\mathbf{x}_1(n),\ldots,\mathbf{x}_{S-1}(n)]$, and $\mathbf{F}^T\mathbf{d}(n)=[d_0(n),d_1(n),\ldots,d_{S-1}(n)]^T$, the SM-SAF minimizes (1) subject to:

$$\mathbf{h}(n+1)\in(\mathbf{\Psi}_{n,0}\cap\mathbf{\Psi}_{n,1}\cap\ldots\cap\mathbf{\Psi}_{n,S-1}) \qquad (33)$$

where:

$$\mathbf{\Psi}_{n,i}=\{\mathbf{h}\in\mathbf{R}^{M}:|d_i(n)-\mathbf{x}_i^T(n)\mathbf{h}|\le\gamma\} \qquad (34)$$

This aim is obtained by an orthogonal projection of the previous estimate of h onto the closest boundary of $\Psi_{n,i}$ in each subband. Doing this, the filter vector update equation for SM-SAF can be stated as:

$$\mathbf{h}(n+1)=\mathbf{h}(n)+\sum_{i=0}^{S-1}\beta_i(n)\frac{\mathbf{x}_i(n)}{\|\mathbf{x}_i(n)\|^2}e_i(n) \qquad (35)$$

where $\mathbf{e}(n)=\mathbf{d}(n)-\mathbf{X}^T(n)\mathbf{h}(n)$, $\mathbf{F}^T\mathbf{e}(n)=[e_0(n),e_1(n),\ldots,e_{S-1}(n)]^T$ and

$$\beta_i(n)=\begin{cases}1-\dfrac{\gamma}{|e_i(n)|} & if\ |e_i(n)|>\gamma\\ 0 & Otherwise\end{cases} \qquad (36)$$

Equation (35) can also be represented as:

$$\mathbf{h}(n+1)=\mathbf{h}(n)+\mathbf{X}(n)\mathbf{W}(n)\beta(n)\mathbf{e}(n) \qquad (37)$$

where $\beta(n)=diag(\beta_0(n),\beta_1(n),\ldots,\beta_{S-1}(n))$, and

$$\mathbf{W}(n)=\mathbf{F}[\delta\mathbf{I}+diag\{diag\{\mathbf{F}^T\mathbf{X}^T(n)\mathbf{X}(n)\mathbf{F}\}\}]^{-1}\mathbf{F}^T.$$

VI. Set-Membership Selective Partial Update Adaptive Filter Algorithms

A. SM-SPU-NLMS Algorithms

The family of SPU-NLMS algorithms are established from Table II. The family of SM-SPU-NLMS algorithms minimizes $\|\mathbf{h}_F(n+1)-\mathbf{h}_F(n)\|^2$ where $F=\{j_1,j_2,\ldots,j_N\}$ denot-e the indices of the N blocks out of K blocks subject to $\mathbf{h}(n+1)\in\Psi_n$. This aim is achieved by following update equation:

$$\mathbf{h}(n+1)=\mathbf{h}(n)+\alpha(n)\mathbf{C}(n)\mathbf{x}(n)e(n) \qquad (38)$$

where $\alpha(n)$ is given by (10), and $\mathbf{C}(n)$ is obtained from TableII.

B. SM-SPU-AP Algorithms

From [9], the SPU-AP minimizes $\|\mathbf{h}_F(n+1)-\mathbf{h}_F(n)\|^2$ subject to $\mathbf{d}(n)=\mathbf{X}^T(n)\mathbf{h}(n)$. This aim is achieved by:

$$\mathbf{h}_F(n+1)=\mathbf{h}_F(n)+\mathbf{X}_F(n)(\mathbf{X}_F^T(n)\mathbf{X}_F(n))^{-1}\mathbf{e}(n) \qquad (39)$$

where:

$$\mathbf{X}_F(n)=[\mathbf{X}_{j1}^T(n),\mathbf{X}_{j2}^T(n),\ldots\mathbf{X}_{jN}^T(n)]^T \qquad (40)$$

is the $NL\times P$ matrix and $\mathbf{X}_i(n)=[\mathbf{x}_i(n),\mathbf{x}_i(n-1),\ldots,\mathbf{x}_i(n-P+1)]$ is the $L\times P$ matrix. The indices of $\mathbf{F}$ are obtained by the following procedure:

1) Compute the following values for $1\le i\le K$:

$$Tr(X_i^T(n)X_i^T(n)) \qquad (41)$$

2) The indices of $\mathbf{F}$ are correspond to N largest values of (41).

The SM-SPU-APA also minimizes $\|\mathbf{h}_F(n+1)-\mathbf{h}_F(n)\|^2$ but subject to $\mathbf{h}\in(\mathbf{\Psi}_n\cap\mathbf{\Psi}_{n-1}\cap\ldots\cap\mathbf{\Psi}_{n-P+1})$. This aim is achieved by following update equation:

$$\mathbf{h}_F(n+1)=\mathbf{h}_F(n)+\mathbf{X}_F(n)(\mathbf{X}_F^T(n)\mathbf{X}_F(n))^{-1}\mathbf{q}\alpha(n)e(n) \qquad (42)$$

C. SM-SPU-TDAF Algorithms

By partitioning the $\mathbf{z}(n)$ into K blocks each of length L, the SM-SPU-TDAF algorithm minimizes $\|\mathbf{z}_F(n+1)-\mathbf{z}_F(n)\|^2$ subject to $\mathbf{z}(n)\in\mathbf{\Phi}_n$. This aim is achieved from (38) where $\mathbf{C}(n)$ is correspond to SPU-TDAF given in Table II.

D. SM-SPU-SAF Algorithms

The SM-SPU-SAF algorithm minimizes $\|\mathbf{h}_F(n+1)-\mathbf{h}_F(n)\|^2$ subject to (33). From (22), the SM-SPU-SAF algorithm is obtained from the following update equation:

$$\mathbf{h}(n+1)=\mathbf{h}(n)+\mathbf{A}\mathbf{X}(n)\mathbf{F}[\mathbf{\Gamma}(n)]^{-1}\mathbf{F}^T\beta(n)\mathbf{e}(n) \qquad (43)$$

VII. Computational Complexity

From [9], the number of multiplications and division for NLMS algorithm is $2M+2$, and 1 division respectively at every iteration. The SPU-NLMS algorithm needs $M+NL+2$ multiplications and 1 division. The number of comparisons in this algorithm is $O(K)+K\log2(N)$ [9]. In the SM-NLMS, the adaptation is related to the condition in (10). This relation determines that when the filter coefficients should be updated at every adaptation. If the condition in (10) always becomes true (which in practice it does not), then the computational complexity of SM-NLMS is $2M+2$ multiplications and 2 divisions which is similar to the complexity of NLMS. But the gains of applying the SM-NLMS algorithm comes through the reduced number of required updates, which cannot be accounted for a priori, and an increased performance as compared to the NLMS algorithm. The computational complexity of SM-SPU-NLMS algorithms is similar to the SPU-NLMS in the worst case. The computational complexity of the TDAF, and SPU-TDAF algorithms is from [12]. The computational complexity of thr SM-TDAF, and SM-SPU-TDAF algorithms is related to the condition in (10). If the condition in (10) always becomes true (which in practice it does not), then the computational complexity of SM-TDAF, and SM-SPU-TDAF is similar to the TDAF, and SPU-TDAF algorithms. For the family of SAF algorithms, the computational complexity is from [15]. Table III summarizes the computational complexity of the algorithms.

TABLE III
COMPUTATIONAL COMPLEXITY OF THE NLMS, TDAF, SAF, SPU-NLMS, SPU-TDAF, SPU-SAF, SM-NLMS, SM-TDAF, SM-SAF, SM-SPU-NLMS, SM-SPU-TDAF, AND SM-SPU-SAF ALGORITHMS

Algorithm	Multiplications	Divisions
NLMS	*2M+2*	1
TDAF	*O(M)+7M+1*	1
SAF	*3M+3SD+1*	1
SPU-NLMS	*M+NL+2*	1
SPU-TDAF	*O(M)+6M+NL+1*	1
SPU-SAF(Based on the first criterion)	*2M+NL+3SD+1*	1
SPU-SAF(Based on the second criterion)	*2M+NL+3SD+2*	2
SM-NLMS	*2M+2*	1
SM-TDAF	*O(M)+7M+1*	2
SM-SAF	*3M+3SD+1*	2
SM-SPU-NLMS	*M+NL+2*	2
SM-SPU-TDAF	*O(M)+6M+NL+1*	2
SM-SSPU-SAF(Based on the first criterion)	*2M+NL+3SD+1*	2
SM-SSPU-SAF(Based on the second criterion)	*2M+NL+3SD+2*	3

VIII. Simulation Results

We demonstrate the performance of the proposed algorithms by several computer simulations in a system identification scenario. In the system identification setup, the unknown systems have 32 and 64 taps and are selected at random. The input signal, $\mathbf{x}(n)$, is a first order autoregressive (AR(1)) signal generated according to:

$$x(n) = \rho x(n-1) + \omega(n) \tag{44}$$

where $\omega(n)$ is a zero mean white Gaussian signal and the parameter ρ was set to *0.9*. The measurement noise, $v(n)$, with $\sigma_v^2 = 10^{-3}$ was added to the noise free desired signal generated through $d(n) = \mathbf{h}_t^T \mathbf{x}(n)$, where $\mathbf{h}_t$ is the true unknown filter vector. The adaptive filter and the unknown filter vector are assumed to have the same number of taps For the TDAF algorithm, a *M*-point Discrete Cosine Transform (DCT) was employed as the orthogonal transform. Also in APA and for *M* = 32, the parameter *P* was set to 3, and for *M* = 64, this parameter was set to 5. In all the simulations, the simulated learning curves were obtained by ensemble averaging over 200 independent trials. For *M* = 32, the number of blocks (*K*) was set to 4 and for *M* = 64, this parameter was set to 8. Also the value of γ was set to $\sqrt{5\sigma_v^2}$ [19].

Figs 2 and 3 show the learning curves of SPU-APA and SM-SPU-APA for *M* = 32. For the SPU-APA, the step-size is set to $\mu = 1$ and $\mu = 0.1$. Fig. 2 shows the results for *N* = 3. As we can see, the SM-SPU-APA has both fast convergence similar to that of SPU-APA and a significantly lower steady-state MSE than ordinary SPU-APA. Furthermore, the average number of updates in SM-SPU-APA was 514 instead of 4000 in SPU-APA. Fig. 3 shows the results for *N* = 4. Again, the good performance of SM-SPU-APA can be seen. In this simulation, the the average numbers of updates was 411 instead of 4000 in SPU-APA. Fig. 4-6 show the results for *M* = 64. For the SPU-APA, the step-size is set to 0.1 and 1. Again, the SM-SPU-APA has both fast convergence similar to that of SPU-APA and a significantly lower steadystate MSE than ordinary SPU-APA. Fig. 4 shows the results for *N* = 6. The average numbers of updates in SM-SPU-APA was 796 instead of 4000 in SPU-APA. Figs. 5 and 6 show the results for *N* = 7, and 8 respectively. Simulation results show the good performance of the SM-SPU-APA. In Fig. 5, the average number of updated was 677 and in Fig. 6, the average number of updates was 622. Table IV summarizes the average number of updates in SM-SPU-APA for different parameters.

Fig. 7-9 show the learning curves of SPU-TDAF and SMSPU- TDAF for *M* = 32. For the SPU-TDAF, the step-size is set to $\mu = 1$ and $\mu = 0.1$. Fig. 2 shows the results for *N* = 2. As we can see, the SM-SPU-TDAF has both fast convergence similar to that of SPU-TDAF and a significantly lower steadystate MSE than ordinary SPU-TDAF. Furthermore, the average numbers of updates in SM-SPU-TDAF was 975 instead of 8000 in SPU-TDAF. Fig. 8 shows the results for *N* = 3. Again, the good performance of SM-SPU-TDAF can be seen. In this simulation, the the average numbers of updates was 777 instead of 8000 in SPU-TDAF. For *N* = 4, the average number of updates was 691. Figs. 10-12 show the results for *M* = 64. For the SPU-TDAF, the step-size is set to 0.1 and 1. Again, the SM-SPU-TDAF has both fast convergence similar to that of SPU-TDAF and a significantly lower steady-state MSE than ordinary SPU-TDAF. Fig. 10 shows the results for *N* = 6. The average numbers of updates in SM-SPU-TDAF was 1189 instead of 8000 in SPU-TDAF. Figs. 11 and 12 show the results for *N* = 7, and 8 respectively. Simulation results show the good performance of the SM-SPU-TDAF algorithm. In Fig. 11, the average number of updated was 1095 and in Fig. 12, the average number of updates was 1033 instead of 8000 in SPU-TDAF algorithm. Table V summarizes the average number of updates in SM-SPU-TDAF for different parameters.

IX. Summary and Conclusions

In this paper, we presented a unified approach to establish the NLMS, the family of APA, TDAF, SAF, and the adaptive filters with selective partial updates. Accordingly, we presented a unified strategy to derivation of SM and SM-SPU adaptive filter algorithms. We demonstrated the good performance of two proposed algorithms which were called SM-SPU-AP and SM-SPU-TDAF algorithms in system identification application.

TABLE IV
THE AVEARAGE NUMBER OF UPDATES IN SM-SPU-AP ALGORITHM

Algorithm	*M=32*	M=64
SM-SPU-APA,*N=3*	*514*	-
SM-SPU-APA,*N=4*	*411*	-
SM-SPU-APA,*N=6*	-	*796*
SM-SPU-APA,*N=7*	-	*677*
SM-SPU-APA,*N=8*	-	*622*

TABLE V
THE AVEARAGE NUMBER OF UPDATES IN SM-SPU-TDAF ALGORITHM

Algorithm	*M=32*	*M=64*
SM-SPU-TDAF,*N=2*	*975*	-
SM-SPU-TDAF,*N=3*	*777*	-
SM-SPU-TDAF,*N=4*	*691*	-
SM-SPU-TDAF,*N=6*	-	*1189*
SM-SPU-TDAF,*N=7*	-	*1095*
SM-SPU-TDAF,*N=8*	-	*1033*

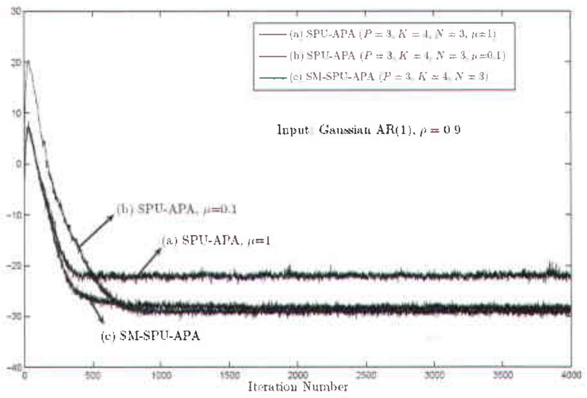

Fig. 2. Learning curves of SPU-APA, and SM-SPU-APA algorithms for $M = 32$, $P = 3$, $K = 4$, and $N = 3$. (input: Gaussian AR(1), ρ=0.9)

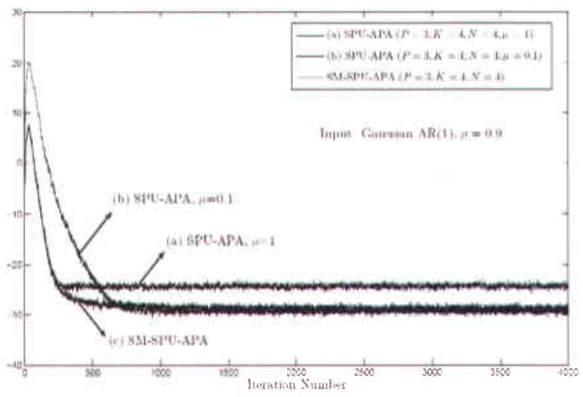

Fig. 3. Learning curves of SPU-APA, and SM-SPU-APA algorithms for $M = 32$, $P = 3$, $K = 4$, and $N = 4$. (input: Gaussian AR(1), ρ=0.9)

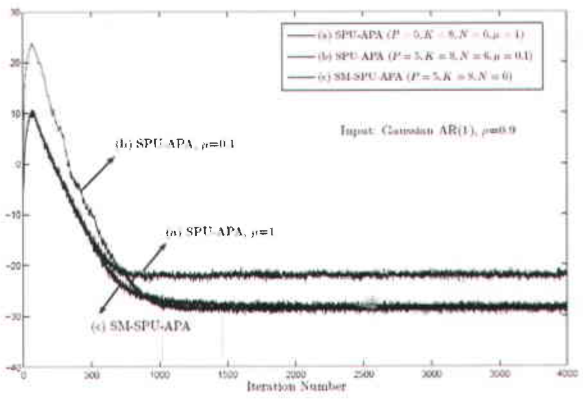

Fig. 4. Learning curves of SPU-APA, and SM-SPU-APA algorithms for $M = 64$, $P = 5$, $K = 8$, and $N = 6$. (input: Gaussian AR(1), ρ=0.9)

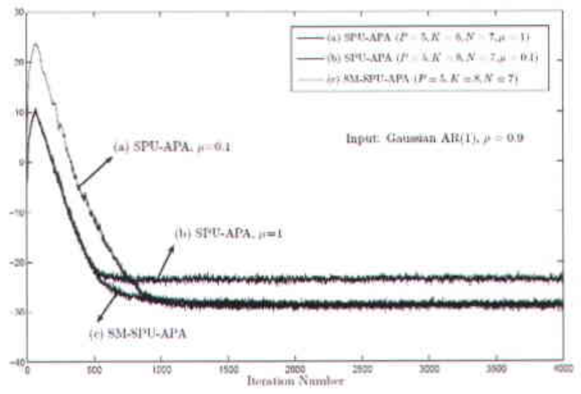

Fig. 5. Learning curves of SPU-APA, and SM-SPU-APA algorithms for $M = 64$, $P = 5$, $K = 8$, and $N = 7$. (input: Gaussian AR(1), ρ=0.9)

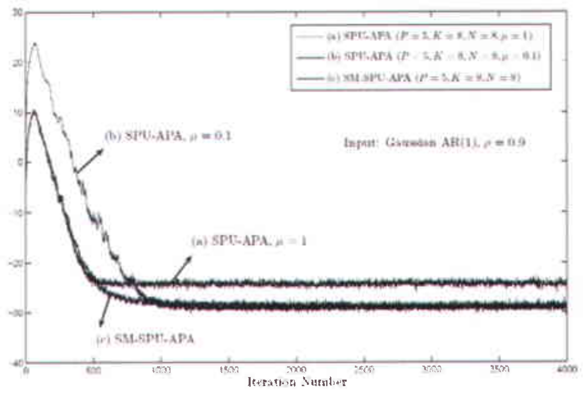

Fig. 6. Learning curves of SPU-APA, and SM-SPU-APA algorithms for $M = 64$, $P = 5$, $K = 8$, and $N = 8$. (input: Gaussian AR(1), ρ=0.9)

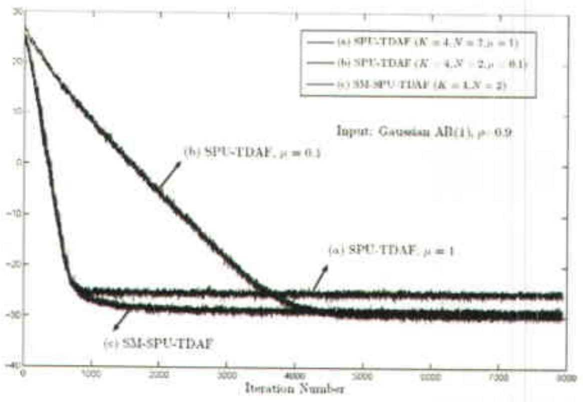

Fig. 7. Learning curves of SPU-TDAF, and SM-SPU-TDAF algorithms for $M = 32$, $K = 4$, and $N = 2$ (input: Gaussian AR(1), ρ=0.9)

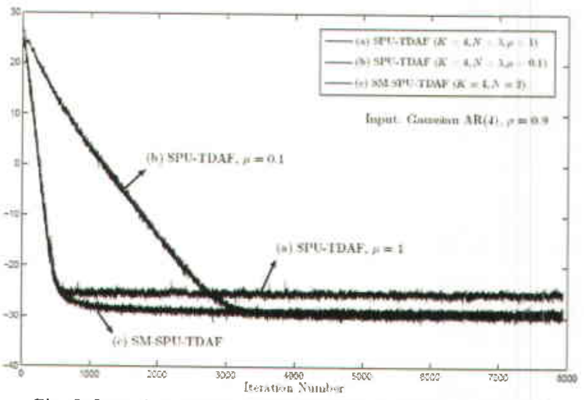

Fig. 8. Learning curves of SPU-TDAF, and SM-SPU-TDAF algorithms for $M = 32$, $K = 4$, and $N = 3$ (input: Gaussian AR(1), ρ=0.9)

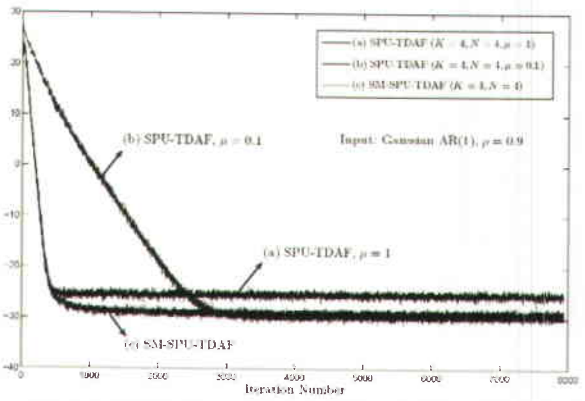

Fig. 9. Learning curves of SPU-TDAF, and SM-SPU-TDAF algorithms for $M = 32$, $K = 4$, and $N = 4$ (input: Gaussian AR(1), ρ=0.9)

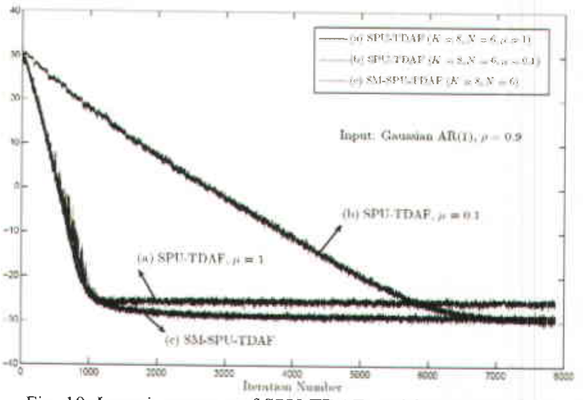

Fig. 10. Learning curves of SPU-TDAF, and SM-SPU-TDAF algorithms for $M = 64$, $K = 8$, and $N = 6$ (input: Gaussian AR(1), ρ=0.9)

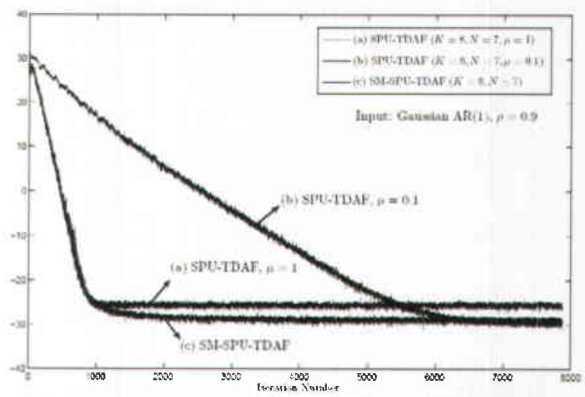

Fig. 11. Learning curves of SPU-TDAF, and SM-SPU-TDAF algorithms for $M = 64$, $K = 8$, and $N = 7$ (input: Gaussian AR(1), ρ=0.9)

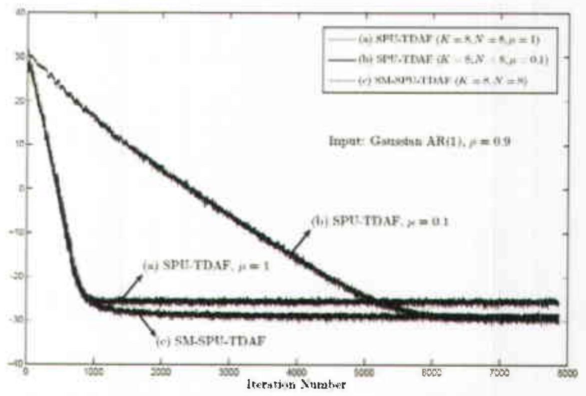

Fig. 12. Learning curves of SPU-TDAF, and SM-SPU-TDAF algorithms for $M = 64$, $K = 8$, and $N = 8$ (input: Gaussian AR(1), ρ=0.9)

REFERENCES

[1] S. Haykin, *Adaptive Filter Theory*. NJ: Prentice-Hall, 4th edition, 2002.

[2] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*, 2nd ed. Kluwer, 2002.

[3] A. H. Sayed, *Fundamentals of Adaptive Filtering*. Wiley, 2003.

[4] ——, *Adaptive Filters*. Wiley, 2008.

[5] S. C. Douglas, "Analysis and implementation of the max- LMS adaptive filter," in *Proc. 29th Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, Oct. 1995, pp. 659–663.

[6] T. Aboulnasr and K. Mayyas, "Selective coefficient update of gradientbased adaptive algorithms," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Munich, Germany, Apr. 1997, pp. 1929–1932.

[7] ——, "Complexity reduction of the NLMS algorithm via selective coefficient update," *IEEE Trans. Signal Processing*, vol. 47, no. 5, pp. 1421–1424, May 1999.

[8] T. Schertler, "Selective block update NLMS type algorithms," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Seattle, WA, May 1998, pp. 1717–1720.

[9] K. Do˘ganc¸ay and O. Tanrıkulu, "Adaptive filtering algorithms with selective partial updates," *IEEE Trans. Circuits, Syst. II: analog and Digital Signal Processing*, vol. 48, no. 8, pp. 762–769, Aug. 2001.

[10] M. S. E. Abadi and J. H. Husøy, "Mean-square performance of the family of adaptive filters with selective partial updates," *Signal Processing*, vol. 88, pp. 2008–2018, 2008.

[11] K. Do˘ganc¸ay, *Partial-Update Adaptive Signal Processing, Design Analysis and implementation*. Academic Press, 2009.

[12] ——, "Complexity considerations for transform-domain adaptive filters," *Signal Processing*, vol. 83, pp. 1177–1192, 2003.

[13] M. S. E. Abadi, V. Mehrdad, and M. Noroozi, “”a family of selective partial update affine projection adaptive filtering algorithms,” *Iranian Journal of Electrical and Electronic Engineering (IJEEE)*, vol. 5, pp. 159–169, 2009.

[14] M. S. E. Abadi and H. Palangi, “Mean-square performance analysis of the family of selective partial update and selective regressor affine projection algorithms,” *Signal Processing*, vol. 90, pp. 197–206, 2010.

[15] M. S. E. Abadi and J. H. Husøy, “Selective partial update and setmembership subband adaptive filters,” *Signal Processing*, vol. 88, pp. 2463–2471, 2008.

[16] S. Gollamudi, S. Nagaraj, S. Kapoor, and Y. F. Huang, “Set-membership filtering and a set-membership normalized LMS algorithm with an adaptive step-size,” *IEEE Signal Processing Letters*, vol. 5, pp. 111– 114, May 1998.

[17] S. Werner and P. S. R. Diniz, “Set-membership affine projection algorithm,” *IEEE Signal Processing Letters*, vol. 8, pp. 231–235, Aug. 2001.

[18] P. S. R. Diniz, R. P. Braga, and S. Werner, “Set-membership affine projection algorithm for echo cancellation,” in *Proc. ISCAS*, Island of Kos, Greece, May 2006, pp. 405–408.

[19] P. S. R. Diniz and S. Werner, “Set-membership binormalized datareusing LMS algorithms,” *IEEE Trans. Signal Processing*, vol. 51, pp. 124–134, Jan. 2003.

[20] S. Werner, M. L. R. de Campos, and P. S. R. Diniz, “Partial-update NLMS algorithms with data-selective updating,” *IEEE Trans. Signal Processing*, vol. 52, pp. 938–949, Apr. 2004.

[21] K. A. Lee and W. S. Gan, “Improving convergence of the NLMS algorithm using constrained subband updates,” *IEEE Signal Processing Letters*, vol. 11, pp. 736–739, 2004.

[22] J. H. Husøy and M. S. E. Abadi, “Unified approach to adaptive filters and their performance,” *IET Signal Processing*, vol. 2, pp. 97–109, 2008.

[23] ——, “A common framework for transient analysis of adaptive filters,” in *Proc. 12th IEEE Mediterranean Electrotechnical Conference*, Dubrovnik, Croatia, May 2004, pp. 265–268.

[24] ——, “Transient analysis of adaptive filters using a general framework,” *Automatika, Journal for Control, Measurement, Electronics, Computing and Communications*, vol. 45, pp. 121–127, 2004.

[25] M. S. E. Abadi and A. M. Far, “A unified approach to steady-state performance analysis of adaptive filters without using the independence assumptions,” *Signal Processing*, vol. 87, pp. 1642–1654, 2007.

[26] S. L. Gay and J. Benesty, *Acoustic Signal Processing for Telecommunication*. Boston, MA: Kluwer, 2000.

[27] J. Apolinario, M. L. Campos, and P. S. R. Diniz, “Convergence analysis of the binormalized data-reusing LMS algorithm,” *IEEE Trans. Signal Processing*, vol. 48, no. 11, pp. 3235–3242, Nov. 2000.

[28] S. S. Pradhan and V. E. Reddy, “A new approach to subband adaptive filtering,” *IEEE Trans. Signal Processing*, vol. 47, pp. 655–664, 1999.

[29] M. de Courville and P. Duhamel, “Adaptive filtering in subbands using a weighted criterion,” *IEEE Trans. Signal Processing*, vol. 46, pp. 2359– 2371, 1998.

[30] S. Werner, M. L. R. de Campos, and P. S. R. Diniz, “Partial-update NLMS algorithms with data-selective updating,” *IEEE Trans. Signal Processing*, vol. 52, no. 4, pp. 938–948, Apr. 2004.

Mohammad Shams Esfand Abadi was born in Tehran, Iran, on September 18, 1978. He received the B.S. degree in Electrical Engineering from Mazandaran University, Mazandaran, Iran, and the M.S. degree in Electrical Engineering from Tarbiat Modares University, Tehran, Iran in 2000 and 2002, respectively, and the Ph.D. degree in Biomedical Engineering from Tarbiat Modares University, Tehran, Iran in 2007. Since 2004 he has been with the Department of Electrical Engineering, Shahid Rajaee University, Tehran, Iran. During the fall of 2003, spring 2005, and again in the spring of 2007, he was a visiting scholar with the Signal Processing Group at the University of Stavanger, Norway. His research interests include digital filter theory and adaptive signal processing algorithms.

Hamid Palangi received M.Sc. degree in electrical engineering from Sharif University of Technology, 2010 , Tehran, Iran and B.Sc. degree in electrical engineering from Shahid Rajaee university, Tehran, Iran, 2007. His research interests include Statistical and adaptive Signal Processing, Inverse problems in Signal Processing, Information Theory and Coding, Applied Signal Processing (Image/Speech) and Tracking, Positioning and Remote Sensing.