ISSN 0005–1144
ATKAAF 45(3–4), 121–127 (2004)

John Håkon Husøy, Mohammad Shams Esfand Abadi

Transient Analysis of Adaptive Filters Using a General Framework

UDK 621.372.543:004.421
IFAC 4.3.2; 5.8.1

Original scientific paper

Employing a recently introduced framework in which a large number of adaptive filter algorithms can be viewed as special cases, we present a generalized transient analysis. An important implication of this is that while the theoretical analysis is performed for a *generic* filter coefficient update equation the results are directly applicable to a large range of adaptive filter algorithms simply by specifying some parameters of this generic filter coefficient update equation. In particular we point out that theoretical learning curves for the Least Mean Square (LMS), Normalized Least Mean Square (NLMS), the Affine Projection Algorithm (APA) and its relatives, as well as the Recursive Least Squares (RLS) algorithm are obtained as special cases of a general result. Subsequently, the recently introduced Fast Euclidian Direction Search (FEDS) algorithms as well as the Pradhan-Reddy subband adaptive filter (PRSAF) are used as non-trivial examples when we demonstrate the usefulness and versatility of the proposed approach to adaptive filter transient analysis through an experimental evaluation.

Key words: convergence analysis (for adaptive filters), transient analysis (for adaptive filters), subband adaptive filters, euclidean direction search, unified theory for adaptive filters

1 INTRODUCTION

Adaptive filtering is an important subfield of digital signal processing having been actively researched for more than four decades and having important applications such as noise cancellation, system identification, telecommunications channel equalization, and telephony acoustic and network echo cancellation. The various adaptive filtering algorithms that have been developed have traditionally been presented without a unifying theoretical framework: Typically, each adaptive filter algorithm is developed from a particular optimization problem whose iterative or direct minimization gives rise to the various algorithms. This approach obscures the relationships, commonalities and differences, between the numerous adaptive algorithms available today. Also, contributions dealing with performance analysis of adaptive filtering algorithms focus on a particular algorithm, making more or less restrictive assumptions on the input signal. Obviously, a more general framework for the understanding and performance analysis encompassing as many different adaptive algorithms as possible as special cases, while at the same time making as few restrictive assumptions as possible, is highly desirable.

In the case of transient analysis, – or convergence analysis, important recent contributions are the analysis of *data normalized* adaptive algorithms [1] (for example the Normalized Least Mean Square (NLMS) algorithm) and the (family of) Affine Projection Algorithm(s) (APA) [2] where excellent agreement between theoretically obtained results and simulations are obtained. What we propose here is a formalism for the transient analysis based on a *generic* adaptive filter update equation proposed in [3] which was shown to cover LMS, NLMS, APA, and RLS as special cases obtained through parameter selections in the generic filter vector update equation. Here we also show that the recently introduced Fast Euclidian Direction Search (FEDS) algorithm [4] as well as the Pradhan-Reddy subband adaptive filters (PRSAF) [5] fit into the class of algorithms that can be viewed as special cases of the generic update equation of [3]. Based on this new insight we exemplify the power and versatility of the proposed transient analysis approach by demonstrating excellent agreement between theoretical and real learning curves for both the FEDS and PRSAF algorithms.

We have organized our paper as follows: In the following section we present the generic update equation forming the basis of our analysis, and briefly review its origin. In the main section, we concisely formulate and solve the problem of finding a general expression for the learning curve of a generic adaptive filter encompassing many particular, classical as well as modern, adaptive filters as special cases. Although the scope of our analysis is wider than that of [2], the logic of the development closely follows that of [2]. Finally, before concluding and summarizing the paper, we show specific examples of theoretically predicted learning curves for the FEDS and PRSAF algorithms demonstrating excellent agreement between theoretically predicted and real learning curves.

2 THE GENERIC UPDATE EQUATION

The generic filter vector update equation at the center of our analysis can be stated as [3]

$$\underline{h}(n+1)=\underline{h}(n)+\mathbf{C}^{-1}(n)\mathbf{X}(n)\mathbf{W}(n)\underline{e}(n). \qquad (1)$$

We use a notation based on the prototypical adaptive filtering setup shown in Figure 1 and explained in Table 1.

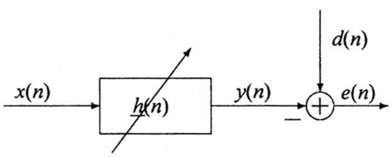

Fig. 1 Prototypical adaptive filter setup

Table 1 **Explanation of notation**

$\underline{h}(n)$	Length M *column* vector of filter coefficients to be adjusted at each time instant n
$\underline{x}(n)$	Length M vector of input signal samples to adaptive filter, $[x(n),x(n-1),\ldots x(n-M+1)]^T$
$\underline{e}(n)$	Length L vector of error samples, $[e(n),e(n-1),\ldots e(n-L+1)]^T$
$\mathbf{X}(n)$	$M\times L$ signal matrix whose columns are given by $\underline{x}(n),\underline{x}(n-1),\ldots \underline{x}(n-L+1)$
$\mathbf{W}(n)$	$L\times L$ symmetric *weighting* matrix
$\mathbf{C}^{-1}(n)$	$M\times M$ inverse *splitting* matrix

Note that all vectors are columns unless explicitly transposed through the superscript ^{T} notation. The latter two matrices of Table 1 and the rationale for Equation 1 enter the picture through the reasoning below. For more details, please refer to [3].

An important goal for all adaptive filters is the rapid convergence to an accurate solution of the Wiener-Hopf equation in a stationary environment. The Wiener-Hopf equation is

$$\mathbf{R}_{xx}\underline{h}=\underline{r}_{xd}, \qquad (2)$$

where $\underline{h}$ is the $M\times 1$ vector of filter coefficients to be determined, $\mathbf{R}_{xx}$ is the autocorrelation matrix of the filter input signal,

$$\mathbf{R}_{xx}=E\{\underline{x}(n)\underline{x}^T(n)\},$$

and $\underline{r}_{xd}$ is the crosscorrelation vector defined by

$$\underline{r}_{xd}=E\{\underline{x}(n)d(n)\}.$$

Since we cannot expect exact knowledge of $\mathbf{R}_{xx}$ and $\underline{r}_{xd}$ of Equation 2, and because it is reasonable to assume those quantities to be time dependent, it makes sense to formulate the adaptive filtering problem as the problem of finding the time dependent solution, $\underline{h}(n)$, to

$$\hat{\mathbf{R}}_{xx}(n)\underline{h}(n)=\hat{\underline{r}}_{xd}(n), \qquad (3)$$

where $\hat{\mathbf{R}}_{xx}(n)$ and $\hat{\underline{r}}_{xd}(n)$ denote estimates of the correlation quantities of Equation 2. Defining the $M\times L$ data matrix

$$\mathbf{X}(n)=[\underline{x}(n),\underline{x}(n-1),\ldots,\underline{x}(n-L+1)], \qquad (4)$$

and given some $L\times L$ full rank symmetric weighting matrix $\mathbf{W}(n)$, we could reasonably state the estimated Wiener-Hopf equation, Equation 3, as

$$\mathbf{X}(n)\mathbf{W}(n)\mathbf{X}^T(n)\underline{h}(n)=\mathbf{X}(n)\mathbf{W}(n)\underline{d}(n), \qquad (5)$$

where $\underline{d}(n)$ is an $L\times 1$ vector of desired signal samples. We notice that if $\mathbf{W}(n)=\mathbf{I}$, where $\mathbf{I}$ is the identity matrix, the estimates used are standard sample estimates of the correlation quantities involved. The larger the value of L is selected, the better estimates we would expect. Selecting $\mathbf{W}(n)$ different from the identity matrix makes it possible to use weighted estimates of the correlation quantities. For the case when $\mathbf{W}(n)=[\mathbf{X}^T(n)\mathbf{X}(n)]^{-1}$, or some function of this quantity, it is common to refer to the associated estimates as data normalized estimates.

Applying a *stationary iterative linear equation solver* [6] to Equaton 5 entails a *splitting* of the coefficient matrix $\mathbf{X}(n)\mathbf{W}(n)\mathbf{X}^T(n)$:

$$\mathbf{X}(n)\mathbf{W}(n)\mathbf{X}^T(n)=\mathbf{C}(n)-[\mathbf{C}(n)-\mathbf{X}(n)\mathbf{W}(n)\mathbf{X}^T(n)], \qquad (6)$$

where $\mathbf{C}(n)$ is some full rank $M\times M$ splitting matrix. Furthermore, performing only one iteration according to the splitting above for each time index, n, we have our generic update equation, Equation 1, when we make use of the fact that $\underline{e}(n)=\underline{d}(n)-\mathbf{X}^T(n)\underline{h}(n)$. Based on the above we identified in [3] four special cases of Equation 1, given by specific choices of $L,\mathbf{W}(n)$, and $\mathbf{C}^{-1}(n)$ corresponding to the LMS, NLMS, APA, and RLS algorithms. The particular choices and their corresponding algorithms are summarized as the top four lines in Table 2. In the table μ is a suitably selected constant, and $\tilde{\mathbf{X}}(n)$ is as $\mathbf{X}(n)$ of Equation 4, but with horizontal dimension $K>M$ (rather than L). The last two entries in the table will be explained later. Note also that we have identified parameters that may be selected as variants of the ones tabulated. In addition we have indicated what type of stationary iterative linear equation solver the splittings corresponds to. It is interesting to note that the most common adaptive filtering algorithms can be interpreted as some sort of Richardson iteration [6], the simplest of all iterative linear equation solvers, applied to a particular estimated Wiener-Hopf equation.

Table 2 **Correspondence between special cases of Eq. 1 and various adaptive filtering algorithms**

L	$\mathbf{W}(n)$	$\mathbf{C}^{-1}(n)$	Algorithm
1	1	$\mu\mathbf{I}$ (Richardson iter.)	LMS
1	$\|\underline{x}(n)\|^{-2}$ (variants)	$\mu\mathbf{I}$ (Richardson iter.)	NLMS
$1<L<M$	$[\mathbf{X}^T(n)\mathbf{X}(n)]^{-1}$ (variants)	$\mu\mathbf{I}$ (Richardson iter.)	APA
1	1	$[\tilde{X}^T(n)\tilde{X}(n)]^{-1}$ (Precond. Richardson)	RLS
$L>M$	$\mathbf{I}$	$i_{j(n)}\ \|\tilde{\underline{x}}_{j(n)}(n)\|^{-2}$ (Partial Gauss-Seidel) (variants)	FEDS-1 (FEDS-P)
K ($\mathbf{X}(kN)$)	$\mathbf{F}\alpha(kN)\mathbf{F}^T$ ($\mathbf{F}$ dependent)	$\mu\mathbf{I}$ (Richardson iter.)	PRSAF

We close this section by stressing the main point: A remarkable number of adaptive filter algorithms can be formulated as special cases of the generic update equation, Equation 1. The table above is not necessarily exhaustive. It is not inconceivable that other combinations of L, $\mathbf{W}(n)$, and $\mathbf{C}^{-1}(n)$ will give rise to new adaptive algorithms exhibiting favorable performance/complexity tradeoffs. As we shall see in the next section, *all adaptive algorithms* that can be viewed as special cases of the generic update equation can, under certain assumptions, be given a common convergence analysis. Needless to say, this is rather nice.

3 GENERAL TRANSIENT ANALYSIS

The technicalities of the transient analysis of this section is closely related to the one presented in [2], the main distinction being that we carry out our analysis using the more general update equation, Equation 1, rather than the update equation for the APA algorithm(s). Also the assumptions made in [2] carry over to the present analysis with appropriate adjustments. Given the technical similarity to the work of [2], we leave out some of the details, and focus on the *results* and their significance.

The learning curve of an adaptive filter algorithm is defined by the evolution of the expected squared apriori error with time n, i.e. as $E\{e_a^2(n)\}$, where the apriori error is defined as

$$e_a(n)=\underline{x}^T(n)[\underline{h}_t-\underline{h}(n)]. \tag{7}$$

$\underline{h}_t$ is the unknown filter vector we are trying to estimate. Defining $\underline{\varepsilon}(n)=\underline{h}_t-h(n)$, we have $e_a(n)= =\underline{x}^T(n)\underline{\varepsilon}(n)$. This implies that

$$E\{e_a^2(n)\}=E\{\underline{\varepsilon}^T(n)\underline{x}(n)\underline{x}^T(n)\underline{\varepsilon}(n)\}. \tag{8}$$

Employing the common *independence assumption* [7] and defining the $\mathbf{A}$-weighted norm for some arbitrary vector $\underline{t}$ as $\|\underline{t}\|^2_{\mathbf{A}}=\underline{t}^T\mathbf{A}\underline{t}$, we have

$$E\{e_a^2(n)\}=E\{\underline{\varepsilon}^T(n)\,\mathbf{R}_{xx}\underline{\varepsilon}(n)\}=E\{\|\underline{\varepsilon}(n)\|^2_{\mathbf{R}_{xx}}\}, \tag{9}$$

where again the definition of the autocorrelation matrix $\mathbf{R}_{xx}=E\{\underline{x}(n)\underline{x}^T(n)\}$ has been used. Thus, to find the learning curve, we need to find $E\{\|\underline{\varepsilon}(n)\|^2_{\mathbf{R}_{xx}}\}$ as a function of n.

Indeed we can find a recursion for $E\{\|\underline{\varepsilon}(n)\|^2_{\mathbf{\Sigma}}\}$, where $\mathbf{\Sigma}$ is some arbitrary square symmetric matrix of dimension commensurate with that of $\underline{\varepsilon}(n)$. Assuming a model for the desired signal, $d(n)$, given by

$$d(n)=\underline{x}^T(n)\underline{h}_t+v(n), \tag{10}$$

which we prefer to express as

$$\underline{d}(n)=\mathbf{X}^T(n)\underline{h}_t+\underline{v}(n), \tag{10}$$

where $\underline{v}(n)$ is measurement noise assumed to be independent of the input signal matrix $\mathbf{X}^T(n)$, we can proceed on a rather laborious derivation pretty much along the lines of [2], but based on Equation 1 rather than on the APA recursion. The final result is

$$E\{\|\underline{\varepsilon}(n+1)\|^2_{\mathbf{\Sigma}}\}=E\{\|\underline{\varepsilon}(n)\|^2_{\mathbf{\Sigma}'}\}+E\{\underline{v}^T(n)\mathbf{X}^{\mathbf{\Sigma}}(n)\underline{v}(n)\}, \tag{12}$$

where

$$\begin{aligned}\mathbf{\Sigma}'=\mathbf{\Sigma}-\mathbf{\Sigma}E\{\mathbf{C}^{-1}(n)\mathbf{X}(n)\mathbf{W}(n)\mathbf{X}^T(n)\}-\\ -E\{\mathbf{X}(n)\mathbf{W}(n)\mathbf{X}^T(n)\mathbf{C}^{-T}(n)\}\mathbf{\Sigma}+\\ +E\{\mathbf{X}(n)\mathbf{X}^{\mathbf{\Sigma}}(n)\mathbf{X}^T(n)\},\end{aligned} \tag{13}$$

and

$$\mathbf{X}^{\mathbf{\Sigma}}(n)=\mathbf{W}(n)\mathbf{X}^T(n)\mathbf{\Sigma}\mathbf{C}^{-1}(n)\mathbf{X}(n)\mathbf{W}(n). \tag{14}$$

The *form* of our expressions exactly match those presented in [2] for APA. The difference is only in the definition of the various quantities involved. From this observation, we can rely directly on the results of [2] to establish

$$E\{\|\underline{\varepsilon}(n+1)\|^2_{\underline{\sigma}}\}=E\{\|\underline{\varepsilon}(n)\|^2_{\mathbf{G}\underline{\sigma}}\}+E\{\underline{v}^T(n)\mathbf{X}^{\mathbf{\Sigma}}(n)\underline{v}(n)\}, \tag{15}$$

where

$$\underline{\sigma}=\mathrm{vec}(\mathbf{\Sigma}) \qquad \mathbf{\Sigma}=vec(\underline{\sigma}) \tag{16}$$

is to be interpreted as (left portion) $\underline{\sigma}$ is the vector of columns of $\mathbf{\Sigma}$ stacked under each other, and (right portion) $\mathbf{\Sigma}$ is the the matrix found by taking equal length sub-vectors of $\underline{\sigma}$ and putting them beside each other. Furthermore, for notational convenience the notation $E\{\|\underline{\varepsilon}(n+1)\|^2_{\underline{\sigma}}\}$ is to be interpreted as the $\mathbf{\Sigma}$-weighted norm of $\underline{\varepsilon}(n+1)$, i.e. as $E\{\|\underline{\varepsilon}(n+1)\|^2_{\mathbf{\Sigma}}\}$. Finally, the matrix $\mathbf{G}$ is an $M^2 \times M^2$ matrix given by

$$\mathbf{G} = \mathbf{I} - E\{\mathbf{Q}(n)\} \otimes \mathbf{I} - \mathbf{I} \otimes E\{\mathbf{Q}(n)\} + E\{\mathbf{Q}(n) \otimes \mathbf{Q}(n)\}, \tag{17}$$

where the first identity matrix has dimension $M^2 \times M^2$, the other two identities are of dimension $M \times M$, $\otimes$ denotes the Kronecker product, and $\mathbf{Q}(n)$ is the $M \times M$ matrix given by

$$\mathbf{Q}(n) = \mathbf{X}(n)\mathbf{W}(n)\mathbf{X}^T(n)\mathbf{C}^{-T}(n). \tag{18}$$

The second term of the right hand side of Equation 15, can, once again following the same line of thought as in [2], be written as

$$E\{\underline{v}^T(n)\mathbf{X}^{\mathbf{\Sigma}}(n)\underline{v}(n)\} = \sigma_v^2 \underline{\sigma}^T \underline{\Upsilon}, \tag{19}$$

where $\sigma_v^2 = E\{v^2(n)\}$ and

$$\underline{\Upsilon} = vec(E\{[\mathbf{C}^{-1}(n)\mathbf{X}(n)\mathbf{W}^2(n)\mathbf{X}^T(n)\mathbf{C}^{-T}(n)]\}), \tag{20}$$

giving the recursion as

$$E\{\|\underline{\varepsilon}(n+1)\|^2_{\underline{\sigma}}\} = E\{\|\underline{\varepsilon}(n)\|^2_{\mathbf{G}\underline{\sigma}}\} + \sigma_v^2 \underline{\sigma}^T \underline{\Upsilon}. \tag{21}$$

Focusing again on the learning curve, we substitute $\mathbf{R}_{xx}$ for $\mathbf{\Sigma}$, define $\underline{r}_{xx} = vec(\mathbf{R}_{xx})$, and find

$$E\{e_a^2(n)\} = E\{\|\underline{\varepsilon}(n+1)\|^2_{\underline{r}_{xx}}\} =$$

$$= E\{\|\underline{\varepsilon}(0)\|^2_{G^n \underline{r}_{xx}}\} + \sigma_v^2 \underline{\Upsilon}^T\{\mathbf{I} + \mathbf{G} + \cdots \mathbf{G}^{n-1}\}\underline{r}_{xx}, \tag{22}$$

This expression is easy to compute recursively once we have estimates for $\mathbf{G}$ and $\mathbf{R}_{xx}$. Such estimates are easily obtained from a *single realization* of the signals involved in the adaptive filter.

What we now have is a tool to predict the transient behavior of various adaptive algorithms. Depending on the particular algorithm we wish to study, all we have to do is to specify the quantities L, $\mathbf{W}(n)$, and $\mathbf{C}^{-1}(n)$ of Table 2. A large number of experiments covering all algorithms of Table 2 has been performed. As two nontrivial examples of results obtained, we employ the theory for the prediction of learning curves for the FEDS and the PRSAF algorithms.

4 APPLICATION TO THE FEDS ALGORITHM

The FEDS algorithm was originally formulated as a simplified conjugate gradient adaptive filter in which the search directions were restricted to the Euclidian directions $g_i = [0,\ldots,0,1,0,\ldots,0]^T$, where the one appears in the i-th position. The directions are sequenced through $i = 0,1,\ldots,M-1$ after which time we start again with $i = 0$. At each sample time, n, the filter vector in only one direction is updated. This means that for each sample interval, only one of the filter coefficients are updated. It has later been shown [8, 9] that this algorithm can be interpreted as the application of one Gauss-Seidel iteration to the normal equation

$$\mathbf{X}(n)\mathbf{X}^T(n)\underline{h} = \mathbf{X}(n)\underline{d}(n) \tag{23}$$

at time n. This Gauss-Seidel iteration is [9]

$$h_i^{(new)} = \underline{h}_i^{(prev)} + \frac{\tilde{\underline{x}}_i^T(n)}{\|\tilde{\underline{x}}_i(n)\|^2}\{\underline{d}(n) - \mathbf{X}^T(n)\underline{h}^{(prev)}\}, \tag{24}$$

where $\tilde{\underline{x}}_i(n)$ is the i-th column of $\mathbf{X}^T(n)$. Defining $\mathbf{i}_j$ as the $M \times M$ matrix with a 1 in position (j,j) and zeros in all other positions, we realize that $\mathbf{i}_j\mathbf{A}$, where $\mathbf{A}$ is some arbitrary matrix of appropriate dimensions, is the matrix with the j-th row equal to the j-th row of $\mathbf{A}$, but with zeros in all other positions. Thus we may refer to $\mathbf{i}_j$ as a row-picking operator. With this, a little thought should reveal that rather than expressing the update of Equation 24 for single components of the filter vector, we can formulate it for the whole filter vector as

$$\underline{h}^{(new)} = \underline{h}^{(prev)} + \frac{1}{\|\tilde{\underline{x}}_i(n)\|^2}\mathbf{i}_i\mathbf{X}(n)[\underline{d}(n) - \mathbf{X}^T(n)\underline{h}^{(prev)}], \tag{25}$$

Realizing that we cycle through i as indicated above, we might simply identify the index of the filter coefficient to be updated at time n an integer function of n, say $j(n)$. This function is given by n counted modulo M, or $j(n) = n \otimes M$, where $\otimes$ is the modulo operator. Denoting the new and previous $\underline{h}$ vectors as $\underline{h}(n+1)$ and $\underline{h}(n)$, respectively, and identifying $[\underline{d}(n) - \mathbf{X}(n)\underline{h}(n)]$ as $\underline{e}(n)$, we have the recursion as

$$\underline{h}(n+1) = \underline{h}(n) + \frac{1}{\|\tilde{\underline{x}}_j(n)\|^2}\mathbf{i}_{j(n)}\mathbf{X}(n)\underline{e}(n). \tag{26}$$

Defining $C^{-1}_{(FEDS)}(n) = \frac{1}{\|\tilde{\underline{x}}_j(n)\|^2}\mathbf{i}_{j(n)}$, we immediately realize that this recursion fits into the form of Equation 1. Note that there is no reason to worry about the $C^{-1}_{(FEDS)}(n)$ itself not being invertible! We can modify the algorithm by allowing more than one coefficient update at each time instant giving better performance at the expense of a somewhat higher computational cost. Fortunately, the *additional* coefficient updates are considerably cheaper than the first [9]. Depending on the number of filter coefficient updates performed at each time instant n, we denote this number by P, we shall refer the algorithm as FEDS-P. For $P > 1$, we get a somewhat more involved expression for $C^{-1}_{(FEDS)}(n)$.

From the above, we realize that the transient analysis of the previous section is directly applicable. In a system identification setup we applied an input signal $x(n)$, generated through

$$x(n) = \rho x(n-1) + w(n) \text{ with } \rho = 0.9 \text{ and } \rho = 0.5,$$

i.e. a highly colored and a somewhat colored signal. In one simulation the amplitudes of $w(n)$ had a Gaussian distribution, in the other, a uniform. The unknown filters $\underline{h}_t$ are selected as random length 8 vectors. The assumed filter length of the FEDS was also set to 8 and the window length was set to $L = 32$. Measurement noise, $v(n)$, with $\sigma_v^2 = 10^{-3}$ was added to the noise free desired signal generated through $d(n) = \underline{h}_t^T \underline{x}(n)$. The results of our experiments are shown in Figure 2 and Figure 3.

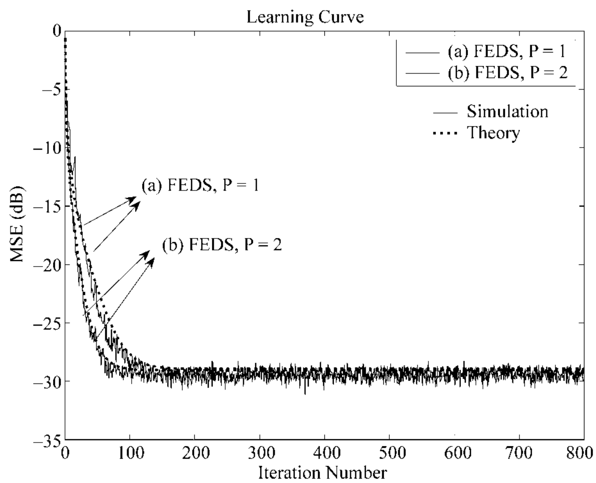

Fig. 2 Simulated and theoretical learning curves for FEDS-P for various P values. L = 32. The input signal was generated according to x(n) = 0.9x(n – 1) + w(n) with w(n) Gaussian

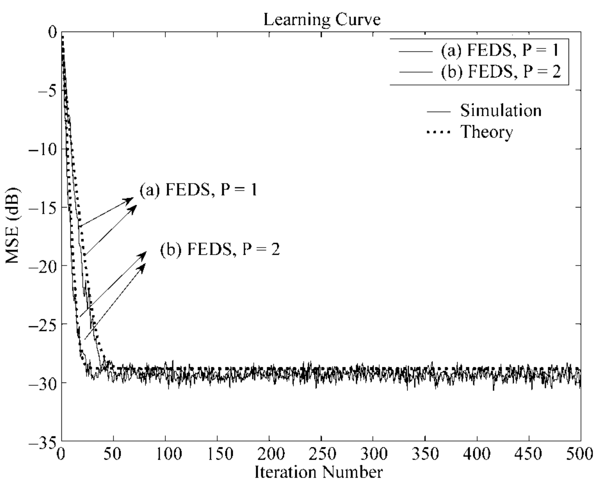

Fig. 3 Simulated and theoretical learning curves for FEDS-P for various P values. L = 32. The input signal was generated according to x(n) = 0.5x(n – 1) + w(n) with w(n) uniformly distributed

As can be seen the agreement between the theoretical learning curves and the learning curves obtained by averaging over 200 independent realizations of the experiment is striking except for the case of $P = 1$, where some discrepancies are observed.

5 APPLICATION TO SUBBAND ADAPTIVE FILTERS

An important class of *subband adaptive filters* is the so called *Pradhan-Reddy Subband Adaptive Filter* (PRSAF) [5], the structure of which is shown in Figure 4. Such filters are described in detail in [5, 10, 11]. PRSAF has received considerable attention in the literature, and it is interesting to note that the adaptive filter algorithms in [5, 10, 11], while derived from different points of view, are the same. To make a long story short, the PRSAF minimizes a weighted sum of the expected squared subband errrors [5] resulting in an update algorithm formulated in terms of sample-by-sample updates, at the sample rate in the subband signals, of the *polyphase components* of the full band adaptive filter $\underline{h}(n)$. This update of the polyphase components can, after some tedious but in principle simple derivations, be shown to imply an update of the equivalent full band filters given by[1]:

$$\underline{h}(k+1) = \underline{h}(k) + \mu \mathbf{X}(kN)\mathbf{F}\alpha(kN)\mathbf{F}^T e(kN), \quad (27)$$

where

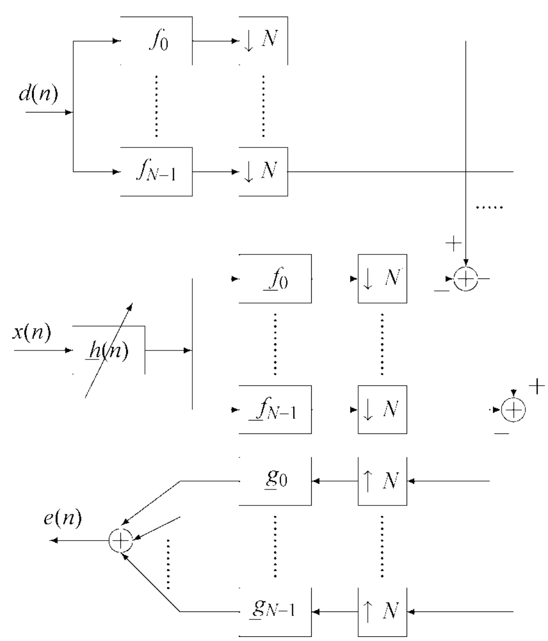

Fig. 4 Structure of the Pradhan-Reddy subband adaptive filter

[1] We note that the Eq. 27 corresponds directly to the formulation given in Eq. 8 of [10].

$$F = [\underline{f}_0, \underline{f}_1, \ldots, \underline{f}_{N-1}], \quad (28)$$

is the $K \times N$ matrix[2)] whose columns are the unit pulse responses of the channel filters of the analysis filter bank of Figure 4,

$$\mathbf{X}(kN) = [\underline{x}(kN), \underline{x}(kN-1), \ldots, \underline{x}(kN-K+1)], \quad (29)$$

and

$$\underline{e}(kN) = \underline{d}(kN) - \mathbf{X}^T(kN)\underline{h}(k). \quad (30)$$

The $N \times N$ matrix $\alpha(kN)$ is a diagonal matrix with elements $\|\mathbf{X}(kN)\mathbf{f}_i\|^{-2}$, $i = 0, 1, \ldots, N-1$ on the diagonal. Note that the terms $\|\mathbf{X}(kN)\mathbf{f}_i\|^2$ are the signal energies of N subband samples in channel no. i of the analysis filter bank. Finally, we point out that this is a *block adaptive* algorithm, i.e. one filter vector update is performed each time N new samples have entered the system.

Based on the above and comparing Equation 27 to Equation 1, – which we recall was identified as an iterative solution strategy for Equation 5, we immediately observe that the PRSAF update can be interpreted as an iterative solution strategy applied to the weighted Wiener-Hopf-type equation

$$\mathbf{X}(kN)\mathbf{F}\alpha(kN)\mathbf{F}^T\mathbf{X}^T(kN)\underline{h}(k) = \mathbf{X}(kN)\mathbf{F}\alpha(kN)\mathbf{F}^T\underline{d}(kN). \quad (31)$$

Furthermore, since we see that the PRSAF fits into the general framework of Section 2, Equation 1 with $\mathbf{C}^{-1}(n)$, $\mathbf{W}(n)$ and $\mathbf{X}(n)$ as given by the last entry in Table 2, is a compact description of the PRSAF. Also, we realize that the convergence speed is determined by the eigenvalue spread of $\mathbf{R_F} = E\{\mathbf{X}(kN)\mathbf{F}\alpha(kN)\mathbf{F}^T\mathbf{X}^T(kN)\}$ suggesting that a design procedure for the filterbank, $\mathbf{F}$, involving the minimization of the eigenvalue spread of $\mathbf{R_F}$ is conceivable. Finally, as will be demonstrated below, the convergence results of Section 3 are directly applicable to the PRSAF family of adaptive filters.

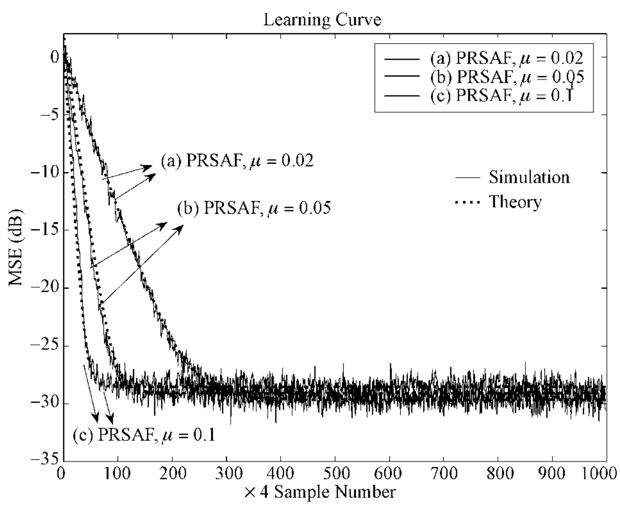

Fig. 5 Simulated and theoretical learning curves for a four band PRSAF with various μ-values and an input signal generated according to x(n) = 0.9x(n – 1) + w(n) with w(n) Gaussian

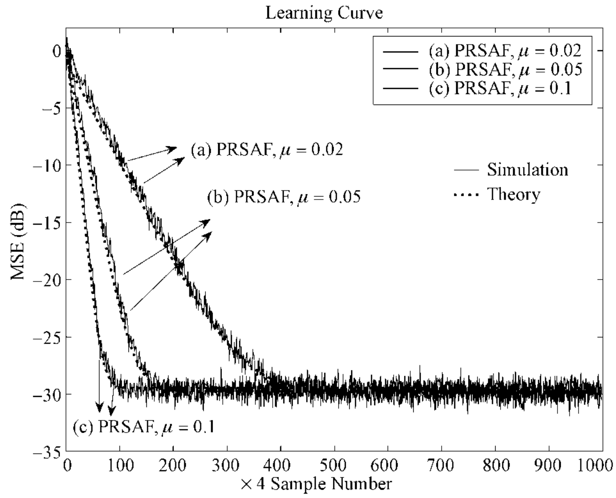

Fig. 6 Simulated and theoretical learning curves for a four band PRSAF with various μ-values and an input signal generated according to x(n) = 0.5x(n – 1) + w(n) with w(n) uniformly distributed

We have performed an experimental evaluation of the applicability of the general convergence theory of Section 3 to PRSAF adaptive filters: We used a system identification setup identical to the one described in the previous section. The filter bank used was the four subband Extended Lapped Transform (ELT) of Malvar [12]. Computed learning curves (using results of Section 3) and simulated learning curves obtained by averaging over 200 independent runs are shown in Figure 5 and Figure 6 for various choices of μ. As is evident we have excellent agreement between simulated learning curves and learning curves predicted by our theory.

6 CONCLUSIONS

Making use of 1) A generic update equation giving many different classes of adaptive filters through simple parameter selection [3] and 2) A line of thought for formulating the central issue in transient analysis [1, 2] previously applied to restricted classes of adaptive filtering algorithms, we have provided a theory allowing us to derive general results for the transient behavior for all algorithms conforming to Equation 1. We demonstrated the usefulness of this theory by showing its excellent ability to predict the performance of the recently introduced FEDS and PRSAF algorithms.

2) K is the length of the channel filters of the analysis filter bank.

REFERENCES

[1] T. Y. Al-Naffouri, A. H. Sayed, **Transient Analysis of Data-normalized Adaptive Filters**. IEEE Trans. Signal Processing, vol. 51, pp. 639–652, Mar. 2003.

[2] H.-C. Shin, A. H. Sayed, **Mean-square Performance of a Family of Affine Projection Algorithms**. IEEE Trans. Signal Processing, vol. 52, pp. 90–102, Jan. 2004.

[3] J. H. Husøy, **A Unified Framework for Adaptive Filtering**. in Proc. NORSIG, Bergen, Norway, Oct. 2003, CD-ROM, ISBN 82-993158-5-9.

[4] G. F. Xu, T. Bose, J. Schroeder, **The Euclidian Direction Search Algorithm for Adaptive Filtering**. In Proc. ISCAS, Orlando, Fl., USA, May 1999, pp. 146–149 (Vol. III).

[5] S. S. Pradhan, V. E. Reddy, **A New Approach to Subband Adaptive Filtering**. IEEE Trans. Signal Processing, vol. 47, pp. 655–664, Mar. 1999.

[6] Y. Saad, **Iterative Methods for Sparse Linear Systems**. PWS Publishing, 1996.

[7] S. Haykin, **Adaptive Filter Theory**, Fourth ed. Upper Saddle River, NJ, USA: Prentice Hall, 2002.

[8] T. Bose, G. F. Xu, **The Euclidian Direction Search Algorithm in Adaptive Filtering**. IEICE Trans. Fundamentals, vol. E85-A, no. 3, pp. 532–539, Mar. 2002.

[9] J. H. Husøy, **RAMP: An Adaptive Filter with Links to Matching Pursuits and Iterative Linear Equation Solvers**. In Proc. ISCAS, Bangkok, Thailand, May 2003, pp. IV-381–384.

[10] M. de Courville, P. Duhamel, **Adaptive Filtering in Subbands Using a Weighted Criterion**. IEEE Trans. Signal Processing, vol. 46, pp. 2359–2371, Sept. 1998.

[11] K. A. Lee, W. S. Gan, **Improving Convergence of the NLMS Algorithm Using Constrained Subband Updates**. IEEE Signal Processing Letters, vol. 11, pp. 736–739, Sept. 2004.

[12] H. Malvar, **Signal Processing with Lapped Transforms**. Artech House, 1992.

Analiza prijelazne pojave adaptivnih filtara primjenom općeg radnog okvira. U radu se predstavlja poopćena analiza prijelaznih pojava adaptivnih filtara, koja se zasniva na primjeni nedavno predstavljenog radnog okvira koji velik broj raznih algoritama adaptivnih filtara promatra kao specijalne slučajeve. Važna posljedica toga je da su rezultati, iako se teoretska analiza provodi na generičkoj jednadžbi za osvježavanje koeficijenta filtra, izravno primjenjivi na razne algoritme adaptivnih filtara jednostavnom specificikacijom nekih parametara generičke jednadžbe za osvježavanje koeficijenata filtra. Posebno se naglašava da su teoretske krivulje učenja za algoritam najmanjih kvadrata (LMS), normalizirani algoritam najmanjih kvadrata (NLMS), afini projekcijski algoritam (APA) i njemu srodnih algoritama, kao i za rekurzivni algoritam najmanjih kvadrata (RLS) dobivene kao posebni slučajevi poopćenog rješenja. Potom se nedavno predstavljeni algoritmi brze euklidske usmjerene pretrage (FEDS) te Pradhan-Reddy pojasni adaptivni filtar (PRSAF) koriste kao netrivijalni primjeri za dokazivanje korisnosti i univerzalnosti predloženog pristupa analizi prijelaznih pojava adaptivnih filtara kroz eksperimentalnu evaluaciju.

Ključne riječi: analiza konvergencije (za adaptivne filtre), analiza prijelaznih pojava (za adaptivne filtre), pojasni adaptivni filtri, euklidske usmjerene pretrage, objedinjena teorija adaptivnih filtara

AUTHORS' ADDRESSES

John Håkon Husøy*
University of Stavanger
Department of Electrical and Computer Engineering,
N-4036 Stavanger, Norway

Mohammad Shams Esfand Abadi
Tarbiat Modares University,
Department of Electrical Engineering
Tehran, Iran

* Corresponding author
E-mail: **john.h.husoy@uis.no** (John Håkon Husøy).

Received: 2004–11–30