A STREAMLINED APPROACH TO ADAPTIVE FILTERS

John Håkon Husøy

University of Stavanger, Department of Electrical and Computer Engineering,
N-4036 Stavanger, Norway (email: John.H.Husoy@uis.no)

ABSTRACT

In this paper we present a streamlined framework for adaptive filters within which all major adaptive filter algorithms can be seen as special cases. The framework involves three ingredients: 1) A preconditioned Wiener Hopf Equation, 2) Its simplest possible iterative solution through the Richardson iteration, and 3) An estimation strategy for the autocorrelation matrix, the cross correlation vector and a preconditioning matrix. This results in a unified adaptive filter theory characterized by simplicity, elegance and economy of ideas suitable for an educational setting in which the similarities and differences between the many different adaptive filter algorithms are stressed.

1. INTRODUCTION

Adaptive filter theory is an important area in digital signal processing with many important applications having been researched for more than four decades. No doubt, adaptive filtering can be considered a mature subject.

All adaptive filter algorithms share the same common goals: 1) Rapid convergence to an accurate solution of the Wiener-Hopf equation in a stationary environment, 2) Good tracking of the time varying Wiener solution in non stationary environments, and 3) Small filter coefficient deviations from the Wiener solution in a stationary environment after convergence. All these objectives shall be satisfied with algorithms characterized by the lowest possible computational complexity. Nevertheless, and in spite of the commonality in goals, the theory of adaptive filters is characterized by a multitude of algorithms whose derivations, both as originally presented and as presented in contemporary graduate level textbooks, rely on a large number ideas that are often perceived by students as somewhat unrelated. Typically, each adaptive filter algorithm is developed from a particular optimization problem whose iterative or direct minimization gives rise to the algorithm. This approach obscures the relationships, commonalities and differences, between the numerous adaptive algorithms available today.

From an educational point of view, a more coherent approach relying on a minimum set of ideas directly related to the objective of finding/tracking the Wiener solution, having significant intuitive appeal, and clearly revealing similarities and differences between the algorithms, would be highly desirable. Such a coherent approach will be presented in this paper. In the fall of 2005 a well received graduate level course on adaptive filter theory adopting this coherent approach was offered at the University of Stavanger.

The main ingredients, – to be presented in sequence below, in our coherent approach to adaptive filtering are: 1) A Preconditioned Wiener Hopf Equation (PCWH), 2) A simple iteration for the solution of this equation, and 3) Reasonable and intuitively meaningful estimation strategies for the random quantities involved in the iteration. As we will see, all common adaptive filters are derived quite economically.

2. THE PRECONDITIONED WIENER HOPF EQUATION AND ITS ITERATIVE SOLUTION

The Wiener-Hopf equation is given by [1]:

$$\mathbf{R}_{xx}\underline{h}_t = \underline{r}_{xd}, \tag{1}$$

where $\underline{h}_t$ is the $M \times 1$ vector of filter coefficients constituting what we refer to as the true Wiener solution, $\mathbf{R}_{xx}$ is the autocorrelation matrix of the filter input signal, $\mathbf{R}_{xx} = E\{\underline{x}(n)\underline{x}^T(n)\}$. $E\{\}$ is the expectation operator, and

$$\underline{x}(n) = [x(n), x(n-1), \ldots, x(n-M+1)]^T \tag{2}$$

is the column vector of random variables corresponding to signal $x(n)$ at various time instants. $\underline{r}_{xd}$ is the cross correlation vector defined by $\underline{r}_{xd} = E\{\underline{x}(n)d(n)\}$[1]. $d(n)$ is commonly referred to as the desired signal.

Applying Richardson's method [2], the simplest of all stationary iterative linear equation solvers [3], to Eq. 1 we get the iteration

$$\underline{h}(n+1) = \underline{h}(n) + \mu[\underline{r}_{xd} - \mathbf{R}_{xx}\underline{h}(n)], \tag{3}$$

where μ is some suitably chosen constant. Note that at this point, the index n is just an iteration index not necessarily related to any signal time index. Defining the filter coefficient deviation at iteration n as $\varepsilon(n) = h_t - \underline{h}(n)$, the iteration is expressed as

$$\underline{\varepsilon}(n+1) = (\mathbf{I} - \mu\mathbf{R}_{xx})\underline{\varepsilon}(n). \tag{4}$$

We remark that this is the type of recursion analyzed in standard treatments of the behavior of the expected coefficient deviation in the Least Mean Square (LMS) algorithm. From these treatments and also from elementary numerical linear algebra it is well known that that convergence is assured when $0 < \mu < 2/\lambda_{max}$, where λ_{max} is the maximum eigenvalue of $\mathbf{R}_{xx}$. Also, convergence slows down as the eigenvalue spread, or the ratio $\frac{\lambda_{max}}{\lambda_{min}}$, increases. The main problem can be seen as the lack of free parameters in the iteration enabling us to exercise control over the convergence speed. For this purpose, we employ the *preconditioning* paradigm from numerical linear algebra [3].

The Preconditioned Wiener Hopf Equation (PCWH) can be stated as

$$\mathbf{C}\mathbf{R}_{xx}\underline{h}_t = \mathbf{C}\underline{r}_{xd}, \tag{5}$$

where $\mathbf{C}$ is some invertible matrix called the *preconditioner*. Obviously the Wiener Hopf equation and its preconditioned version have the same solution. Applying Richardson's method to Eq. 5, we get

$$\underline{h}(n+1) = \underline{h}(n) + \mu\mathbf{C}[\underline{r}_{xd} - \mathbf{R}_{xx}\underline{h}(n)], \tag{6}$$

which, when formulated in terms of the coefficient deviation, gives

$$\underline{\varepsilon}(n+1) = (\mathbf{I} - \mu\mathbf{C}\mathbf{R}_{xx})\underline{\varepsilon}(n). \tag{7}$$

Now $\mathbf{C}\mathbf{R}_{xx}$ plays the same role in Eq. 7 as $\mathbf{R}_{xx}$ does in Eq. 4. Thus, selecting $\mathbf{C}$ as an approximate inverse of $\mathbf{R}_{xx}$, we can lower the eigenvalue spread significantly, and consequently improve the convergence speed dramatically relative to the case when no preconditioner is employed. Of course, the introduction of the preconditioner should not increase the computational demands unduly. The preconditioning paradigm has enjoyed great popularity recently among numerical analysts working on the iterative solution of large sets of linear equations [4]. In the next section we demonstrate the relevance of preconditioning to adaptive filtering.

[1] As is common, we do not use any notation to distinguish the cases when $\underline{x}(n)$ is to be interpreted as a random vector and when it is to be interpreted as a vector of signal samples.

3. FROM ITERATIVE EQUATION SOLVERS TO ADAPTIVE FILTERS

Standard textbook derivations of the LMS algorithm applies the method of steepest descent to $E\{e^2(n)\}$ resulting in the same recursion as stated in Eq. 3 above. The LMS algorithm is obtained by substituting instantaneous estimates for the involved auto- and cross-correlations: $\mathbf{R}_{xx} \rightarrow \underline{x}(n)\underline{x}^T(n)$ and $\underline{r}_{xx} \rightarrow \underline{x}(n)d(n)$. From this, it is not a giant intellectual leap to take the Richardson iteration for the preconditioned Wiener Hopf equation, Eq. 6, as the starting point and substituting reasonable estimates for $\mathbf{R}_{xx}, \mathbf{C}$ and $\underline{r}_{xd}$. Denoting these estimates by $\mathbf{R}_{xx}(n), \mathbf{C}(n)$ and $\underline{r}_{xd}(n)$ we get

$$\underline{h}(n+1) = \underline{h}(n) + \mu\mathbf{C}(n)[\underline{r}_{xd}(n) - \mathbf{R}_{xx}(n)\underline{h}(n)]. \qquad (8)$$

It is important to realize that *any reasonable estimate* of the mentioned quantities will give us an adaptive filter.

Defining the signal matrix $\mathbf{X}(n)$ as

$$\mathbf{X}(n) = [\underline{x}(n), \underline{x}(n-1), \ldots, \underline{x}(n-L+1)], \qquad (9)$$

and the vector of desired signal samples through

$$\underline{d}(n) = [d(n), d(n-1), \ldots, d(n-L+1)]^T, \qquad (10)$$

it is obvious that

$$E\{\mathbf{X}(n)\mathbf{X}^T(n)\} = L \cdot \mathbf{R}_{xx} \qquad (11)$$

and

$$E\{\mathbf{X}(n)\underline{d}(n)\} = L \cdot \underline{r}_{xd}. \qquad (12)$$

Thus,

$$\mathbf{R}_{xx}(n) = \mathbf{X}(n)\mathbf{X}^T(n) \qquad (13)$$

and

$$\underline{r}_{xd}(n) = \mathbf{X}(n)\underline{d}(n) \qquad (14)$$

are reasonable estimates for $\mathbf{R}_{xx}$ and $\underline{r}_{xx}$[2]. Using these estimates in Eq. 8, we have what we sometimes refer to as our generic adaptive filter:

$$\underline{h}(n+1) = \underline{h}(n) + \mu\mathbf{C}(n)\mathbf{X}(n)\underline{e}(n), \qquad (15)$$

where $\underline{e}(n) = \underline{d}(n) - \mathbf{X}^T(n)\underline{h}(n)$. Setting $L = 1$, we get what has previously been referred to as instantaneous estimates. The selection of $\mathbf{C}(n)$ requires a bit more care and is treated in the next subsection.

4. THREE PRECONDITIONING STRATEGIES

From the above it is evident that the closer $\mathbf{C}(n)$ is to an inverse of an estimated autocorrelation matrix the better. Of course, the estimate $\mathbf{R}_{xx}(n) = \mathbf{X}(n)\mathbf{X}^T(n)$ is not necessarily invertible. In fact for $L < M$ the estimate is rank deficient and definitely not invertible. For $L > M$, $\mathbf{R}_{xx}(n)$ will be invertible for reasonable input signals. We suggest three intuitively reasonable strategies for the selection of $\mathbf{C}(n)$:

Constant $\mathbf{C}(n)$: Setting $\mathbf{C}(n)$ arbitrarily to the identity matrix, we get the LMS algorithm when we set $L = 1$ in the definitions of $\mathbf{R}_{xx}(n)$ and $\underline{r}_{xd}(n)$ above. If we have some a priori knowledge of autocorrelation matrices to be encountered in an application, this information can be employed in the determination of a suitable $\mathbf{C}$. This was successfully explored in [5, 6], where $\mathbf{C}$ was chosen as a circulant matrix in order to preserve the low computational complexity of the standard LMS.

$\mathbf{C}(n)$ related to $\mathbf{R}_{xx}(n)$: Since $\mathbf{R}_{xx}(n)$ may not be directly invertible, we settle for the next best option, the *regularized* inverse of $\mathbf{R}_{xx}(n)$:

$$\mathbf{C}(n) = \{\varepsilon\mathbf{I} + \mathbf{R}_{xx}(n)\}^{-1}, \qquad (16)$$

where ε is some suitably chosen constant.

Independently determined $\mathbf{C}(n)$: There is no law preventing us from forming another estimate of $\mathbf{R}_{xx}$ whose invertibility we ensure by design. For example, $\mathbf{R}_{xx}(n)$ with $L > M$ will do the job. To distinguish the estimate of $\mathbf{R}_{xx}$ used in finding $\mathbf{C}(n)$ from the one used directly in the iteration of Eq. 8, we shall term the former $\tilde{\mathbf{R}}_{xx}(n)$ and also explicitly identify the signal matrices involved through the tilde-notation, i.e. we set $\tilde{\mathbf{R}}_{xx}(n) = \tilde{\mathbf{X}}(n)\tilde{\mathbf{X}}^T(n)$. Another example of such an estimate is the exponentially weighted estimate:

$$\tilde{\mathbf{R}}_{xx}(n) = \sum_{i=0}^{n} \lambda^{n-i}\underline{x}(i)\underline{x}^T(i), \qquad (17)$$

where $0 << \lambda < 1$.

5. EXAMPLES

We have already seen that the simplest possible selection of the estimates involved in Eq. 8: $\mathbf{R}_{xx}(n) = \underline{x}(n)\underline{x}^T(n)$, $\underline{r}_{xd}(n) = \underline{x}(n)d(n)$, and $\mathbf{C}(n) = \mathbf{I}$, gives us the LMS algorithm. In this section we show how intuitively plausible alternative selections for $\mathbf{R}_{xx}(n)$, $\underline{r}_{xd}(n)$, and $\mathbf{C}(n)$ quite expediently yield all major adaptive filter algorithms. The results, clearly indicating similarities and differences between the various algorithms, are collected in Table 1.

5.1 $\mathbf{C}(n)$ directly related to $\mathbf{R}_{xx}(n)$

Substituting $\mathbf{R}_{xx}(n)$ and $\underline{r}_{xd}(n)$ according to Eqs. 13 and 14, selecting the preconditioner according to the second strategy of the previous section, i.e. $\mathbf{C}(n) = \{\varepsilon\mathbf{I} + \mathbf{R}_{xx}(n)\}^{-1}$, in our generic iteration, Eq. 8, we get

$$\underline{h}(n+1) = \underline{h}(n) + \mu\{\varepsilon\mathbf{I} + \mathbf{X}(n)\mathbf{X}^T(n)\}^{-1}\mathbf{X}(n)\underline{e}(n). \qquad (18)$$

We assume $L < M$. Applying the matrix inversion lemma [1], we realize that

$$\{\varepsilon\mathbf{I} + \mathbf{X}(n)\mathbf{X}^T(n)\}^{-1}\mathbf{X}(n) = \mathbf{X}(n)\{\varepsilon\mathbf{I} + \mathbf{X}^T(n)\mathbf{X}(n)\}^{-1}. \qquad (19)$$

Using this, the recursion of Eq. 18 can be stated as

$$\underline{h}(n+1) = \underline{h}(n) + \mu\mathbf{X}(n)\{\varepsilon\mathbf{I} + \mathbf{X}^T(n)\mathbf{X}(n)\}^{-1}\underline{e}(n), \qquad (20)$$

which we immediately recognize as the *Affine Projection Algorithm* (APA) that has been employed successfully in many practical applications. Setting $L = 1$, we establish the *Normalized Least Mean Square* (NLMS) algorithm.

5.2 Independently determined $C(n)$

Using instantaneous estimates, i.e. setting $L = 1$ in $\mathbf{R}_{xx}(n)$ and $\underline{r}_{xd}(n)$ as given by Eqs. 13 and 14, in combination with the third strategy above for selecting $\mathbf{C}(n)$, we get

$$\underline{h}(n+1) = \underline{h}(n) + \tilde{\mathbf{R}}_{xx}^{-1}(n)\underline{x}(n)e(n). \qquad (21)$$

Using the estimate for $\tilde{\mathbf{R}}(n)$ given by Eq. 17, we get the ordinary exponentially weighted RLS algorithm, whereas setting $\tilde{\mathbf{R}}_{xx}(n) = \tilde{\mathbf{X}}(n)\tilde{\mathbf{X}}^T(n)$ results in a sliding window version.

Given an autocorrelation matrix $\mathbf{R}_{xx}$, it is well known that if $\mathbf{U}$ is its eigenvalue matrix, then $\mathbf{U}^T\mathbf{R}_{xx}\mathbf{U}$ is diagonal. It is observed that for many input signals encountered in practice, a suitably chosen fixed orthogonal transform matrix, $\mathbf{T}$, – a prominent example is the Discrete Cosine Transform (DCT) matrix, will *approximately* diagonalize $\mathbf{R}_{xx}$. Thus, we have $\mathbf{R}_{xx} \approx \mathbf{T} \cdot \text{diag}[\mathbf{T}^T\mathbf{R}_{xx}\mathbf{T}] \cdot \mathbf{T}^T$, and consequently

$$\mathbf{R}_{xx}^{-1} \approx \mathbf{T}\{\text{diag}[\mathbf{T}^T\mathbf{R}_{xx}\mathbf{T}]\}^{-1}\mathbf{T}^T. \qquad (22)$$

This directly motivates the following estimated preconditioner:

$$\mathbf{C}(n) = \mathbf{T}\{\text{diag}[\mathbf{T}^T\tilde{\mathbf{X}}(n)\tilde{\mathbf{X}}^T(n)\mathbf{T}]\}^{-1}\mathbf{T}^T, \qquad (23)$$

[2] Actually these are scaled estimates, but the scaling factor is not important here.

where the M nonzero elements of $\text{diag}[\mathbf{T}^T\tilde{\mathbf{X}}(n)\tilde{\mathbf{X}}^T(n)\}\mathbf{T}]$ are seen to be running (scaled) averages of transform coefficient powers. Substituting the preconditioner of Eq. 23, still using $\mathbf{R}_{xx}(n)$ and $\underline{r}_{xd}(n)$ with $L=1$, into Eq. 8, we get

$$\underline{h}(n+1)=\underline{h}(n)+\mu\mathbf{T}\{\text{diag}[\mathbf{T}^T\tilde{\mathbf{X}}(n)\tilde{\mathbf{X}}^T(n)\mathbf{T}]\}^{-1}\mathbf{T}^T\underline{x}(n)e(n). \tag{24}$$

Expressing this in terms of transformed quantities, $\underline{\beta}(n)=\mathbf{T}^T\underline{h}(n)$ and $\underline{x}_T(n)=\mathbf{T}^T\underline{x}(n)$, we get, by premultiplying Eq. 24 by $\mathbf{T}^T$:

$$\underline{\beta}(n+1)=\underline{\beta}(n)+\mu\{\text{diag}[\mathbf{T}^T\tilde{\mathbf{X}}(n)\tilde{\mathbf{X}}^T(n)\mathbf{T}]\}^{-1}\underline{x}_T(n)e(n). \tag{25}$$

Both Eqs. 24 and 25 are what is commonly referred to as *Transform Domain Adaptive Filters* (TDAF).

5.3 An interesting generalization

If we look back at Eqs. 13 and 14 one might ask if more general estimates for $\mathbf{R}_{xx}(n)$ and $\underline{r}_{xx}(n)$ can be formulated. Clearly, $E\{\mathbf{X}(n)\mathbf{T}\mathbf{T}^T\mathbf{X}^T(n)\}=M\cdot\mathbf{R}_{xx}$, when $\mathbf{T}$ is an orthogonal $M\times M$ matrix and the horizontal dimension of $\mathbf{X}(n)$ is chosen to be $L=M$. Let us now accept the premise that the $L\times N$ matrix $\mathbf{F}=[\underline{f}_0,\underline{f}_1,\underline{f}_{N-1}]$, where $\underline{f}_i$ is the unit pulse response of the i'th analysis channel of an N channel orthogonal, critically sampled prefect reconstruction analysis/synthesis filter bank system, can be considered a generalization of a square orthogonal matrix. With this, one might speculate if $E\{\mathbf{X}(n)\mathbf{F}\mathbf{F}^T\mathbf{X}^T(n)\}$ is given by some scalar times $\mathbf{R}_{xx}$. Before proceeding we note that it is common to set the length of the channel filters in filter bank systems equal to an integer times the number of channels, i.e. $L=P\cdot N$. Using the fact that filter bank matrices, as described here, satisfy $\sum_{i=0}^{P-1-l}\mathbf{F}_i\mathbf{F}_{i+l}^T=\delta(l)\mathbf{I}$ for $l=0,1,\ldots,P-1$ where the $\mathbf{F}_i$'s are the P submatrices of size $N\times N$ constituting $\mathbf{F}$ [7], it is straight forward, but quite tedious, to show that indeed $E\{\mathbf{X}(n)\mathbf{F}\mathbf{F}^T\mathbf{X}^T(n)\}=N\cdot\mathbf{R}_{xx}$. Similarly, we can show that $E\{\mathbf{X}(n)\mathbf{F}\mathbf{F}^T\underline{d}(n)\}=N\cdot\underline{r}_{xd}$. Assuming that our estimates are updated every N'th input sample, the relevant estimates for the autocorrelation matrix and the cross correlation vector become

$$\mathbf{R}_{xx}(k)=\mathbf{X}(kN)\mathbf{F}\mathbf{F}^T\mathbf{X}^T(kN), \tag{26}$$

and

$$\underline{r}_{xd}(k)=\mathbf{X}(kN)\mathbf{F}\mathbf{F}^T\underline{d}(kN). \tag{27}$$

The preconditioner of choice according to the strategy of Eq. 16, is

$$\mathbf{C}(k)=\{\varepsilon\mathbf{I}+\mathbf{X}(kN)\mathbf{F}\mathbf{F}^T\mathbf{X}^T(kN)\}^{-1}. \tag{28}$$

Substituting these quantities into our generic iteration, Eq. 8, we get

$$\underline{h}(k+1)=\underline{h}(k)+\mu\{\varepsilon\mathbf{I}+\mathbf{X}(kN)\mathbf{F}\mathbf{F}^T\mathbf{X}^T(kN)\}^{-1}\mathbf{X}(kN)\mathbf{F}\underline{e}_F(k), \tag{29}$$

where $\underline{e}_F(k)=\mathbf{F}^T[\underline{d}(kN)-\mathbf{X}^T(kN)\underline{h}(k)]$. Now, applying the matrix inversion lemma in exactly the same fashion as we did when going from Eq. 18 to Eq. 20, we arrive at

$$\underline{h}(k+1)=\underline{h}(k)+\mu\mathbf{X}(kN)\mathbf{F}\{\varepsilon\mathbf{I}+\mathbf{F}^T\mathbf{X}^T(kN)\mathbf{X}(kN)\mathbf{F}\}^{-1}\underline{e}_F(k). \tag{30}$$

The elements in row j of $\mathbf{F}^T\mathbf{X}^T(kN)$ are identified as M consecutive subband signal samples from subband no. j in the above mentioned filter bank system. Similarly, column i of $\mathbf{X}(kN)\mathbf{F}$ are seen to be consecutive subband signal samples from subband no. i. For well designed filter banks, it is known that sample cross correlations of signals from different subbands are very low [8]. Thus, setting

$$\mathbf{F}^T\mathbf{X}^T(kN)\mathbf{X}(kN)\mathbf{F}\approx\text{diag}\{\mathbf{F}^T\mathbf{X}^T(kN)\mathbf{X}(kN)\mathbf{F}\}, \tag{31}$$

is justifiable. Defining the $N\times N$ matrix $\mathbf{\Lambda}(k)$ by

$$\mathbf{\Lambda}^2(k)=\text{diag}\{\mathbf{F}^T\mathbf{X}^T(kN)\mathbf{X}(kN)\mathbf{F}\}, \tag{32}$$

where we identify the diagonal elements as subband signal energies for the various filter bank channels, our final recursion becomes

$$\underline{h}(k+1)=\underline{h}(k)+\mu\mathbf{X}(kN)\mathbf{F}\{\varepsilon\mathbf{I}+\mathbf{\Lambda}^2(k)\}^{-1}\underline{e}_F(k). \tag{33}$$

This recursion is seen to be identical to the subband adaptive filter of Pradhan and Reddy [9]. This adaptive filter, sometimes referred to as the *Pradhan Reddy Subband Adaptive Filter* (PRSAF), was independently derived also in [10, 11].

5.4 Alternate form of the generic adaptive filter

We have previously referred to Eq. 15 as our generic adaptive filter. When we used the preconditioning strategy of Eq. 16, as we did in the derivation of APA and PRSAF, we employed the matrix inversion lemma, see Eq. 19, in arriving at the final recursion. In the APA derivation we might introduce the matrix

$$\mathbf{W}(n)=\{\varepsilon\mathbf{I}+\mathbf{X}^T(n)\mathbf{X}(n)\}^{-1}, \tag{34}$$

in which case Eq. 19 can be written as

$$\mathbf{C}(n)\mathbf{X}(n)=\mathbf{X}(n)\mathbf{W}(n). \tag{35}$$

From this one might argue that in some cases, a suitable alternate form of the generic adaptive filter of Eq. 15 should be stated as:

$$\underline{h}(n+1)=\underline{h}(n)+\mathbf{X}(n)\mathbf{W}(n)\underline{e}(n). \tag{36}$$

In the PRSAF derivation we could similarly introduce a $\mathbf{W}_F(k)$ matrix

$$\mathbf{W}_F(k)=\{\varepsilon\mathbf{I}+\text{diag}[\mathbf{F}^T\mathbf{X}^T(kN)\mathbf{X}(kN)\mathbf{F}]\}^{-1}, \tag{37}$$

leading, when accepting the diagonal approximation of Eq. 31, to

$$\mathbf{C}(k)\mathbf{X}(kN)\mathbf{F}=\mathbf{X}(kN)\mathbf{F}\mathbf{W}_F(k), \tag{38}$$

which gives the generic adaptive filter in the form

$$\underline{h}(k+1)=\underline{h}(k)+\mathbf{X}(kN)\mathbf{F}\mathbf{W}_F(k)\underline{e}_F(k). \tag{39}$$

This can be considered as a further generalization of Eq. 36.

6. SUMMARY AND CONCLUSION

In this paper we have presented a streamlined framework for adaptive filters. We have seen that all major adaptive filter algorithms: LMS, NLMS, APA, RLS, TDAF, and PRSAF are all easily derived in a unified way. The differences between the various algorithms are clearly identified as differences in the selection of a preconditioner and the way in which the autocorrelation matrix and cross correlation vector estimates are formed. In Table 1, we have collected the results of our developments related to the generic recursion of Eq. 8. A further benefit of our streamlined approach is the possibility of doing performance analysis on our general recursions rather than for each and every algorithm.

REFERENCES

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

[2] L. F. Richardson, "The approximate arithmetical solution by finite differences of physical problems involving differential equations with an application to the stresses to a masonry dam," *Philos. Trans. Roy. Soc. London*, vol. Ser A 210, pp. 307–357, 1910.

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

[4] K. Chen, *Matrix Preconditioning Techniques and Applications*. Cambridge University Press, 2005.

Algorithm	$\mathbf{C}(n)$	$\mathbf{R}_{xx}(n)$	$\underline{r}_{xd}(n)$	L
LMS	$\mathbf{I}$	$\underline{x}(n)\underline{x}^T(n)$	$\underline{x}(n)d(n)$	1
NLMS*	$(([\varepsilon\mathbf{I}+\underline{x}(n)\underline{x}^T(n)]^{-1}))$	$\underline{x}(n)\underline{x}^T(n)$	$\underline{x}(n)d(n)$	1
APA*	$(([\varepsilon\mathbf{I}+\mathbf{X}(n)\mathbf{X}^T(n)]^{-1}))$	$\mathbf{X}(n)\mathbf{X}^T(n)$	$\mathbf{X}(n)\underline{d}(n)$	$1 < L < M$
RLS	$[\tilde{\mathbf{X}}(n)\tilde{\mathbf{X}}^T(n)]^{-1}$ (sliding window) or $[\sum_{i=0}^{n}\lambda^{n-i}\underline{x}(i)\underline{x}^T(i)]^{-1}$ (exp. weighted window)	$\underline{x}(n)\underline{x}^T(n)$	$\underline{x}(n)d(n)$	1
TDAF	$\mathbf{T}\cdot\{\mathrm{diag}[\mathbf{T}^T\tilde{\mathbf{X}}(n)\tilde{\mathbf{X}}^T(n)\mathbf{T}]\}^{-1}\cdot\mathbf{T}^T$ or $\mathbf{T}\cdot\{\mathrm{diag}[\sum_{i=0}^{n}\lambda^{n-i}\mathbf{T}^T\underline{x}(i)\underline{x}^T(i)\mathbf{T}]\}^{-1}\cdot\mathbf{T}^T$	$\underline{x}(n)\underline{x}^T(n)$	$\underline{x}(n)d(n)$	1
PRSAF*	$(([\varepsilon\mathbf{I}+\mathbf{X}(kN)\mathbf{F}\mathbf{F}^T\mathbf{X}^T(kN)]^{-1}))$	$\mathbf{X}(kN)\mathbf{F}\mathbf{F}^T\mathbf{X}^T(kN)$	$\mathbf{X}(kN)\mathbf{F}\mathbf{F}^T\underline{d}(kN)$	$P\cdot N$

Table 1: Table indicating specific choices of the central parameters of an adaptive filter, $\mathbf{R}_{xx}(n)$, $\underline{r}_{xd}(n)$, and $\mathbf{C}(n)$ for all major adaptive filter families.
$*$: For the APA (and NLMS), and PRSAF it may be preferable to state the generic adaptive filter as given in Eqs. 36 or 39, in which cases the the $\mathbf{W}$-matrices are given by $\mathbf{W}(n) = \{\varepsilon\mathbf{I}+\mathbf{X}^T(n)\mathbf{X}(n)\}^{-1}$, and $\mathbf{W}_F(k) = \{\varepsilon\mathbf{I}+\mathrm{diag}[\mathbf{F}^T\mathbf{X}^T(kN)\mathbf{X}(kN)\mathbf{F}]\}^{-1}$, respectively.

[5] J. H. Husøy and M. S. E. Abadi, “A new LMS-type algorithm utilizing approximate a priori knowledge of the input autocorrelation,” in *Proc. Applied Electronics*, (Plzen, Czech Republic), pp. 147 – 150, Sept. 2005.

[6] J. H. Husøy and M. S. E. Abadi, “A novel LMS-type adaptive filter optimized for operation in multiple signal environments,” in *Proc. NORSIG*, (Stavanger, Norway), pp. 117 – 120, Sept. 2005.

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

[8] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs: Prentice Hall, 1993.

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

[10] M. de Courville and 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 and W. S. Gan, “Improving convergence of the NLMS algorithm using constrained subband updates,” *IEEE Signal Processing Letters*, vol. 11, pp. 736–739, Sept. 2004.