A Critical View of Adaptive Filter Theory

John Håkon Husøy

Deptartment of Electrical Engineering and Computer Science
University of Stavanger, N-4036 Stavanger, Norway

john.h.husoy@uis.no

Abstract – **In this paper we offer a "birds eye view" of the field of adaptive filtering with the aim of suggesting a new approach to its teaching. Following a brief survey of the diverse lines of thought underpinning the traditional approach to the presentation of the most common adaptive filter algorithms, we point out that all major families of adaptive filter algorithms can be more easily developed and understood in terms of the well established theory of iterative linear equation solvers. This simplifies both the development and analysis of the most commonly available adaptive filter algorithms and gives additional insight. In particular the commonalities and differences between the various algorithms are more easily understood. This enhanced and complementary understanding, inspired by numerical linear algebra, also paves the way for the conception of new adaptive filter algorithms.**

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.

Most adaptive filter algorithms share the same common goals: 1) Rapid convergence to a good approximation of the solution to 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 of ideas that are often perceived by students as unrelated. This will be clear from our brief survey of the most common approaches to the teaching of modern adaptive filter algorithms as presented below.

In this paper we promote an alternative and unified approach to adaptive filters based on central elements of the theory of iterative linear equation solvers. We will point out that all major adaptive filters can be directly derived from the common starting point of a *preconditioned* Wiener-Hopf equation. As intuitively appealing preconditioners and estimates for the correlation quantities of the Wiener-Hopf equation are selected the various common adaptive filter algorithms are easily derived.

We have organized our paper as follows: In the next section, we establish notation and briefly review historically important approaches to the presentation and development of the Least Mean Square (LMS) algorithm and in its normalized version (NLMS), the Recursive Least Squares (RLS) algorithm and the Affine Projection Algorithm (APA). This serves to illustrate the need for a more coherent and unified approach which we present subsequently. This approach is based on the application of a simple iteration, the Richardson iteration that can be dated back to 1910 [1], to a preconditioned version of the Wiener-Hopf equation. From the presented critical approach to the field of adaptive filtering, it is hoped the the reader gains an appreciation for our coherent and unified alternative approach to this important branch of digital signal processing.

The classical textbook approach

Before briefly reviewing the conceptual underpinnings of the main workhorses in adaptive filter theory, namely the LMS[1], RLS and the APA algorithms, we introduce some notation through the prototypical adaptive filtering setup of Figure 1.

Adaptive filters can be seen as online algorithms adjusting the coefficients of an FIR filter $\underline{h}(k)$, given as a length M *column* vector of filter coefficients, in such a way that the *output signal*, $y(k)$, is a good estimate of the *desired signal*, $d(k)$.

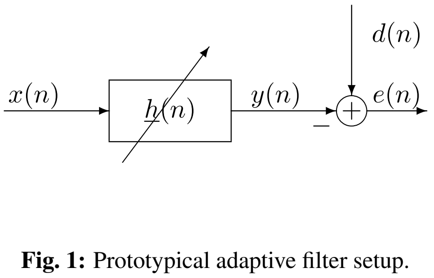

Fig. 1: Prototypical adaptive filter setup.

The LMS algorithms is found by applying the *steepest decent* iterative optimization strategy to the objective function $E\{e^2(k)\}$ [2] , where $E\{\}$ is the expectation

[1] Along with its *normalized* version, the NLMS algorithms.

operator, and $e(k)$ is the output error signal

$$e(k) = d(k) - \underline{h}^T(k)\underline{x}(k). \qquad (1)$$

The signal vector $\underline{x}(k)$ is defined through

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

where T denotes matrix transposition. The gradient of the objective function with respect the the filter vector is given by $\nabla E\{e^2(k)\} = -2\underline{r} + 2\mathbf{R}\underline{h}$, where $\mathbf{R}$ is the auto correlation matrix of the filter input signal, $\mathbf{R} = E\{\underline{x}(k)\underline{x}^T(k)\}$, and $\underline{r}$ is the cross correlation vector defined by $\underline{r} = E\{\underline{x}(k)d(k)\}$. The minimization of our objective function is now performed iteratively: In each iteration we move the current estimate of $\underline{h}$ some amount in the direction of the negative gradient. That is:

$$\underline{h}(k+1) = \underline{h}(k) + \frac{\mu}{2}[-\nabla E\{e^2(k)\}], \qquad (3)$$

giving

$$\underline{h}(k+1) = \underline{h}(k) + \mu[\underline{r} - \mathbf{R}\underline{h}(k)]. \qquad (4)$$

μ is an adaption constant. Substitution of *instantaneous estimates* for the correlation quantities, i.e. letting $\mathbf{R} \rightarrow \underline{x}(k)\underline{x}^T(k)$ and $\underline{r} \rightarrow \underline{x}(k)d(k)$ we get the LMS algorithm [2]:

$$\underline{h}(k+1) = \underline{h}(k) + \mu\underline{x}(k)e(k), \qquad (5)$$

What we have presented above is by far the most common approach to the introduction of LMS adaptive filters to advanced undergraduate or beginning graduate students. This material is often presented along with an introduction to the FIR *Wiener filter* which is found by assuming that the true auto correlation matrix and cross correlation vectors are known or can be found exactly. When the latter assumption about the availability of the correlation quantities is valid the optimization problem is solved directly by setting above mentioned gradient equal to zero. This results in the Wiener-Hopf equation

$$\mathbf{R}\underline{h}_t = \underline{r}, \qquad (6)$$

whose solution gives the FIR Wiener filter. Although the gifted, may be even the average, student may very well conjecture that the LMS algorithm really is an algorithm for iteratively solving the Wiener-Hopf equation while its coefficient matrix, $\mathbf{R}$, and its right hand side, $\underline{r}$, may change continuously, this idea is not developed further in the most commonly used textbooks. This is a pity since it prevents us from conveying, quite easily, an understanding for the connections between the many available adaptive filtering algorithms. Furthermore, this also prevents us from tapping into the large body of knowledge available on iterative schemes for the solution of sets of linear equations, which as we shall point out shortly, is directly applicable to the study of adaptive filters.

Following the above development of the LMS algorithm, most courses on adaptive filters proceed with the development of the NLMS, RLS and APA adaptive filters. Typically, the NLMS algorithm is developed by allowing the adaption constant μ to be substituted with a time variant quantity, $\mu(k)$, which is determined by minimizing $E\{e^2(k+1)\}$ given the update structure of the LMS algorithm as given in (5). Another popular approach, but from a student perspective viewed as quite unrelated, also resulting in the NLMS algorithm solves the constrained optimization problem min$\|\underline{h}(k+1) - \underline{h}(k)\|^2$ with respect to $\underline{h}(k+1)$ subject to the equality constraint $\underline{x}^T(k)\underline{h}(k+1) = d(k)$.

The APA adaptive filter can be developed and/or motivated using geometric arguments. However, more popular is the argument that is based on a generalization of the second approach to the NLMS derivation of the previous paragraph in which the equality constraint is substituted with the *set of* equality constraints $\underline{x}^T(k-n)\underline{h}(k+1) = d(k-n)$ for $n = 0, 1, \ldots, P-1$. The integer P is referred to as the projection order.

Moving on to the RLS algorithm, the students are often left with the impression that in this case we have yet another optimization problem as the starting point of the algorithm development: In contrast to the "statistical perspective" employed in deriving the algorithms above, we now tend to present what we introduce as a "deterministic perspective" through the minimization of the objective function

$$\sum_{i=0}^{k} \lambda^{k-i} e^2(i), \qquad (7)$$

with $e(k)$ as given in (1) above and $0 << \lambda < 1$. Depending on the scope of the adaptive filter course in question, one now typically moves on to the various transform domain and subband domain algorithms. By way of example, considering what is commonly referred to as the Pradhan-Reddy subband adaptive filter (PRSAF) we have the choice of (at least) three quite different and rather complicated arguments [3, 4, 5] giving us the adaptive filter algorithm.

Based on the short review above, almost needless to say, the students' perception of the subject of adaptive filters are sometimes quite confused. This is our motivation for suggesting a change in teaching practice that is based on a numerical linear algebra perspective in which we base ourselves on a 1) *preconditioned* Wiener-Hopf equation, 2) a simple iterative linear equation solver (the Richardson iteration) and 3) the substitution of estimates for the auto and cross correlation quantities, $\mathbf{R}$ and $\underline{r}$.

The unified approach

Preliminaries

The preconditioned Wiener Hopf equation (PCWH) can be stated as

$$\mathbf{C}\mathbf{R}\underline{h}_t = \mathbf{C}\underline{r}, \qquad (8)$$

where $\mathbf{C}$ is some invertible matrix called the *preconditioner*. The Wiener Hopf equation and its preconditioned version have the same solution. Applying Richardson's method, [1], the simplest of all stationary iterative linear equation solvers [6], to (8), we get

$$\underline{h}(k+1) = \underline{h}(k) + \mu\mathbf{C}[\underline{r} - \mathbf{R}\underline{h}(k)], \qquad (9)$$

which, when formulated in terms of the coefficient deviation (equation error), $\underline{\epsilon}(k) = \underline{h}_t - \underline{h}(k)$ gives

$$\underline{\epsilon}(k+1) = (\mathbf{I} - \mu\mathbf{CR})\underline{\epsilon}(k). \qquad (10)$$

It is well known that, in complete analogy with the analysis of the mean performance of the LMS algorithm[2], a large eigenvalue spread for $\mathbf{CR}$ gives slow convergence, whereas a small eigenvalue spread facilitates rapid convergence. Thus, selecting $\mathbf{C}$ as an approximate inverse of $\mathbf{R}$, 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 [9]. In the next section we demonstrate the relevance of preconditioning to adaptive filtering.

From iterative linear equation solvers to adaptive filters

In (9) we might consider substituting the quantities $\mathbf{R}$, $\underline{r}$ and $\mathbf{C}$ with suitable estimates available at the time when the k-th iteration is performed. Doing so, assuming that the estimates are updated based on signal samples up to and including *time* kN, where N is some suitably chosen positive integer, and denoting the estimated quantities by $\mathbf{R}(kN), \underline{r}(kN)$, and $\mathbf{C}(kN)$ we have

$$\underline{h}(k+1) = \underline{h}(k) + \mu\mathbf{C}(kN)[\underline{r}(kN) - \mathbf{R}(kN)\underline{h}(k)]. \qquad (11)$$

Setting $N = 1$ we get sample-by-sample algorithms, i.e. the iteration index and the signal time index coincide, whereas selecting $N > 1$ results in block-based algorithms in which chunks of N samples are input to the algorithm for each coefficient update. It is important to realize that *any reasonable estimate* of the mentioned quantities will give us an adaptive filter. For example, substituting instantaneous estimates for the involved auto- and cross-correlations estimates of (11), with $N = 1$, i.e. $\mathbf{R}(k) \to \underline{x}(k)\underline{x}^T(k)$ and $\underline{r} \to \underline{x}(k)d(k)$ and selecting the identity matrix as the preconditioner, we get

$$\underline{h}(k+1) = \underline{h}(k) + \mu\underline{x}(k)[d(k) - \underline{x}^T(k)\underline{h}(k)], \qquad (12)$$

which is recognized as the LMS algorithm.

Based on the general update of (11) we point out that it is possible to identify general expressions for the estimates $\mathbf{R}(kN), \underline{r}(kN)$ and $\mathbf{C}(kN)$ that are parameterized through a small number of parameters/choices in such a way that the all common adaptive filter algorithms are identified as special cases of (11). For clarity of presentation and economy of space, we avoid full generality and focus on the central ideas. All the details can be found in [10]. It is also interesting to note that this formalism makes it possible to derive general performance results that can be specialized to particular families of adaptive filters through the selection of the mentioned parameters. These important issues are also fully explored in [10].

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

$$\mathbf{X}(k) = [\underline{x}(k), \underline{x}(k-1), \ldots, \underline{x}(k-L+1)], \qquad (13)$$

and the vector of desired signal samples through

$$\underline{d}(k) = [d(k), d(k-1), \ldots, d(k-L+1)]^T, \qquad (14)$$

a fairly general[3] and intuitively appealing form for the (scaled) estimates[4] $\mathbf{R}(kN)$ and $\underline{r}(kN)$ are given by

$$\mathbf{R}(kN) = \mathbf{X}(kN)\mathbf{X}^T(kN), \qquad (15)$$

and

$$\underline{r}(kN) = \mathbf{X}(kN)\underline{d}(kN). \qquad (16)$$

Substituting the estimates of (15) and (16) into (11), we obtain what we refer to as our generalized adaptive filter:

$$\underline{h}(k+1) = \underline{h}(k) + \mu\mathbf{C}(kN)\mathbf{X}(kN)\underline{e}(kN), \qquad (17)$$

where $\underline{e}(kN) = \underline{d}(kN) - \mathbf{X}^T(kN)\underline{h}(k)$. With this, we are ready for the exploration of possible choices for $\mathbf{C}(kN)$.

Three preconditioning strategies

From what we have said so far it is evident that a preconditioner being a good approximation to the inverse of the autocorrelation matrix results in a rapidly converging iterative solution strategy for the Wiener-Hopf equation. In an adaptive filtering environment the exact autocorrelation matrix is unavailable, we can only form estimates of this matrix. These estimates may however be utilized in the formation of intuitively plausible estimated preconditioners, $\mathbf{C}(kN)$, for use in our generalized adaptive filter, (17). Below, we identify three approaches to the selection of $\mathbf{C}(kN)$.

1. **Constant $\mathbf{C}(kN)$:** If we have some prior knowledge of typical autocorrelation matrices to be encountered in an application, this information can be employed in the determination of a suitable fixed preconditioner $\mathbf{C}$. This was successfully explored in [11], where $\mathbf{C}$ was chosen as a sparse and highly structured matrix in order to preserve the low computational complexity of the standard LMS algorithm. We have also generalized this to normalized algorithms, see [12, 13]. We emphasize that these algorithms are a direct consequence of our numerical linear algebra perspective involving preconditioners. Without this perspective, it would be hard to conceive a line of thought that would lead to these novel adaptive filter algorithms. Of course, the LMS algorithm employs this preconditioning strategy in the sense that the preconditioning matrix is set to the identity matrix.

[2] See any of [7, 2, 8].

[3] Though not the most general form, see [10] for details.

[4] Since the same scaling factor is implied for both the estimates of $\mathbf{R}$ and $\underline{r}$ this scaling factor plays no role in the algorithm development. In the following we shall consequently refer to estimates even though they may be scaled.

2. **$\mathbf{C}(kN)$ related to $\mathbf{R}(kN)$:** Since $\mathbf{R}(kN)$ may not be directly invertible, we cannot use its inverse as a preconditioner. However, we might settle for the next best option, the *regularized* inverse of $\mathbf{R}(kN)$:

$$\begin{aligned}\mathbf{C}(kN) &= \{\epsilon\mathbf{I} + \mathbf{R}(kN)\}^{-1} \\ &= \{\epsilon\mathbf{I} + \mathbf{X}(kN)\mathbf{X}^T(kN)\}^{-1}, \qquad (18)\end{aligned}$$

where ϵ is some suitably chosen constant.

3. **Independently determined $\mathbf{C}(kN)$:** Instead of using $\mathbf{R}(kN)$ as the basis for the selection of $\mathbf{C}(kN)$, we could form another estimate of the autocorrelation matrix, denoted by $\tilde{\mathbf{R}}(kN)$, whose invertibility we assure by design. An obvious choice for the preconditioner in this case would be $\mathbf{C}(kN) = \tilde{\mathbf{R}}^{-1}(kN)$. An example of such an autocorrelation matrix estimate is

$$\tilde{\mathbf{R}}(kN) = \tilde{\mathbf{X}}(kN)\tilde{\mathbf{X}}^T(kN), \qquad (19)$$

where the signal matrix, $\tilde{\mathbf{X}}(kN)$, has the same structure as $\mathbf{X}(kN)$, defined in (13), but with horizontal dimension exceeding the vertical dimension M so as to ensure invertibility of $\tilde{\mathbf{R}}(kN)$ for reasonable input signals, $x(k)$. Another example of such an estimate is the exponentially weighted estimate:

$$\tilde{\mathbf{R}}_{xx}(kN) = \sum_{i=0}^{kN} \lambda^{kN-i}\underline{x}(i)\underline{x}^T(i), \qquad (20)$$

where $0 << \lambda < 1$.

Based on the generic iteration of (17) and one of these preconditioning strategies, all important families of adaptive filter algorithms can be derived with a minimum of effort. As an example we provide below a short and simple, yet the complete, argument leading to the APA adaptive filter.

Using the preconditioner of (18) and substituting this into (17) we get

$$\begin{aligned}&\underline{h}(k+1) = \underline{h}(k) + \\ &\mu\{\epsilon\mathbf{I} + \mathbf{X}(kN)\mathbf{X}^T(kN)\}^{-1}\mathbf{X}(kN)\underline{e}(kN), \quad (21)\end{aligned}$$

which in view of the *matrix inversion lemma* [2] can be written as

$$\begin{aligned}&\underline{h}(k+1) = \underline{h}(k) + \\ &\mu\mathbf{X}(kN)\{\epsilon\mathbf{I} + \mathbf{X}^T(kN)\mathbf{X}(kN)\}^{-1}\underline{e}(kN), \quad (22)\end{aligned}$$

which is immediately recognized as the ϵ-APA algorithm [8] when we select $N = 1$. For $L = 1$ this is the NLMS algorithm. We also mention that setting $\epsilon = 0$, and $L = 2$ corresponds to the binormalized data-reusing LMS (BNDR-LMS) adaptive filter algorithm of [14].

Similar short and simple, but yet complete derivations can be made for most other families of modern adaptive filters. Here we present a summary of the results of such a collection of algorithm derivations in Table 1. The parameters characterizing each algorithm should be interpreted with reference to (17) or its alternate, but equivalent, form

$$\underline{h}(k+1) = \underline{h}(k) + \mu\mathbf{X}_F(kN)\mathbf{W}(kN)\underline{e}_F(kN), \quad (23)$$

where $\mathbf{W}(kN)$ is a weighting matrix that is directly related to the preconditioner $\mathbf{C}(kN)$. More details can be found in [10]. The key message here is that there indeed is a systematic line of thought along which modern adaptive filters can be developed in a unified fashion. We have had success with this approach in the teaching of our graduate course on adaptive filters at the University of Stavanger. A nice experience has been the assignment of student projects associated with preconditioning strategy no. 1 based on which students have been able to come up with novel algorithms.

Summary and conclusion

The main issue in this paper has been the advocation of a new approach to the teaching of adaptive filters based on the use of simple iterative linear equation solvers applied to a preconditioned Wiener-Hopf equation. We have shown that with this approach a consistent and unified derivation of the major adaptive filter algorithms, as well as novel algorithms, can be found. Our experience of connecting the various algorithms together via (17), (23) and Tabe 1 enhances the students' understanding of the important field of adaptive filters. Furthermore, it turns out that when dealing with issues of performance, we can do general performance analyses taking (17) or (23) as a starting point. This saves time and enable us to focus on the implications of the results rather than on the technicalities involved in the classical approach in which each and every adaptive filter algorithms is subjected to its own algorithms specific performance analysis.

1. REFERENCES

[1] 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.

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

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

[4] 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.

[5] 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, Sep. 2004.

Algorithm	**C/W**	K	P	N	**F**
LMS (**C**)	$\mathbf{I}$	1	1	1	1
NLMS (**W**)	$[\epsilon\mathbf{I} + \|\underline{x}(k)\|^2]^{-1}$	1	1	1	1
APA (**W**)	$[\epsilon\mathbf{I} + \mathbf{X}^T(k)\mathbf{X}(k)]^{-1}$	$1 < K < M$	1	1	$\mathbf{I}$
PRSAF (**W**)	$[\epsilon\mathbf{I} + \text{diag}\{\mathbf{F}^T\mathbf{X}^T(kN)\mathbf{X}(kN)\mathbf{F}\}]^{-1}$	$1 < K < M$	$P \geq 1$	$N = K$	$\mathbf{F}$
RLS (**C**)	$[\tilde{\mathbf{X}}(k)\tilde{\mathbf{X}}^T(k)]^{-1}$ (sliding window) or $[\sum_{i=0}^{k} \lambda^{k-i}\underline{x}(i)\underline{x}^T(i)]^{-1}$ (exp. weighted window)	1	1	1	1
TDAF (**C**)	$\mathbf{T} \cdot \{\text{diag}[\mathbf{T}^T\tilde{\mathbf{X}}(k)\tilde{\mathbf{X}}^T(k)\mathbf{T}]\}^{-1} \cdot \mathbf{T}^T$ or $\mathbf{T} \cdot \{\text{diag}[\sum_{i=0}^{k} \lambda^{k-i}\mathbf{T}^T\underline{x}(i)\underline{x}^T(i)\mathbf{T}]\}^{-1} \cdot \mathbf{T}^T$	1	1	1	1

Table 1: Tabular identification of the the most common families of adaptive filters withe the parameters of (17) and (23) specified. Note that we here use a somewhat more general formulation than the one used in the text in order to include the Pradhan-Reddy subband adaptive filter (PRSAF) and the transform domain adaptive filter (TDAF). Thus, the adaptive filter algorithms can be described through either $\underline{h}(k+1) = \underline{h}(k) + \mu\mathbf{C}(kN)\mathbf{X}_F(kN)\underline{e}_F(kN)$, or $\underline{h}(k+1) = \underline{h}(k) + \mu\mathbf{X}_F(kN)\mathbf{W}(kN)\underline{e}_F(kN)$, where $\mathbf{X}_F(kN) = \mathbf{X}(kN)\mathbf{F}, \underline{d}_F(kN) = \mathbf{F}^T\underline{d}(kN), \underline{e}_F(kN) = \mathbf{F}^T\underline{e}(kN)$, and $\underline{e}(kN) = \underline{d}(kN) - \mathbf{X}^T(kN)\underline{h}(k)$, and where $\mathbf{F}$ is a matrix whose columns are the unit pulse responses of the analysis part an N channel perfect reconstruction orthogonal filter bank [15]. $\mathbf{T}$ is an orthogonal tranform, for example the Discrete Cosine Transform (DCT). Specification of the mathematical forms of $\mathbf{C}(kN)$ or $\mathbf{W}(kN)$ along with the $P \cdot K \times K$ matrix $\mathbf{F}$ and the block size N uniquely describe an adaptive filter algorithm. Note that in the column identifying algorithm names, we indicate whether $\mathbf{C}(kN)$ or $\mathbf{W}(kN)$ is specified. In the $\mathbf{F}$ column, an $\mathbf{F}$ indicates the use of some well conceived filter bank matrix with little channel overlap.

[6] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed. SIAM, 2003.

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

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

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

[10] J. H. Husøy and M. S. E. Abadi, “A unified approach to adaptive filters and their performances,” *IET Signal Processing*, vol. 2, pp. 97 – 109, Jun. 2008, doi: 10.1049/iet-spr:20070077.

[11] J. H. Husøy, “A preconditioned LMS adaptive filter,” *ECTI Trans. on Electrical Eng., Eelectronics, and Communications*, vol. 5, pp. 3 – 9, Feb. 2007.

[12] ——, “A circulantly preconditioned NLMS-type adaptive filter,” in *Proc. 17th Intl. Conf. Radioelektronika*, Brno, Czech Republic, 2007, pp. 141 – 145.

[13] J. H. Husøy and M. S. E. Abadi, “A family of flexible NLMS-type adaptive filter algorithms,” in *Proc. of Sixth Intl. Conf. on Information, Communications and Signal Processing*, Singapore, Dec. 2007, cD-ROM proceedings, available through IeeeXplore.

[14] J. Apolinario, M. L. R. Campos, and P. S. R. Diniz, “Convergence analysis of the binormalized datareusing LMS algorithm,” *IEEE Trans. Signal Processing*, vol. 48, pp. 3235–3242, Nov. 2000.

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