A Preconditioned LMS adaptive filter

John Håkon Husøy¹, Non-member

ABSTRACT

Inspired by the commonalities and differences between the Least Mean Square (LMS) and RLS/Newton algorithms, we propose a new adaptive filter algorithm, the Preconditioned LMS (PLMS) algorithm, which is optimized with respect to approximate a priori knowledge of the input autocorrelation signal. PLMS can result in significantly improved convergence speed, with negligible complexity increase, relative to the LMS algorithm.

1. INTRODUCTION

Adaptive filtering is important in a variety of applications. The “workhorse” of most applications involving adaptive filters is the Least Mean Square (LMS) algorithm and its many variants. If computational complexity is less of an issue, Recursive Least Squares (RLS) adaptive filtering algorithms can be employed for improved convergence performance [1]. It is well known that when the eigenvalue spread of the autocorrelation matrix of the adaptive filter input signal is large (highly colored input signal), the LMS algorithm converges extremely slowly. On the other hand, the more computationally demanding RLS algorithm has good convergence performance irrespective of the characteristics of the autocorrelation matrix.

A fundamental aim in many adaptive filter research efforts is the conception of algorithms alleviating the poor convergence behavior of the LMS or the normalized LMS (NLMS) algorithms for highly colored inputs. As is evident from the major textbooks on the subject [1, 2] many solutions have been proposed. Common to most of them is the aim of making the adaptive filter’s convergence behavior more or less independent of the autocorrelation properties of the input signal. Examples of such algorithms are 1) The family of Affine Projection Algorithms (APA) which, as the projection order increases, can be seen as an algorithms transitioning from the NLMS towards the RLS algorithm, 2) Transform Domain Adaptive Filters (TDAF) utilizing the ability of well conceived orthogonal transforms in approximately diagonalizing common autocorrelation matrices, and 3) Subband Adaptive Filters (SAF) of the type published in [3–5]. While the aim of making the convergence behavior

of adaptive filters independent of the input autocorrelation properties is a reasonable one, we here look at the adaptive filtering problem from a somewhat different point of view.

For white input signals, it is well known that the LMS algorithm has as good convergence behavior as the RLS algorithm. For such inputs, we might thus say that the LMS algorithm has optimum performance. When the LMS algorithm is used in applications where the input is known to be colored, the adaptive filter designer implicitly settles for suboptimum performance which, depending on the characteristics of the application at hand, may be acceptable. Given some more or less precise information about the expected autocorrelation properties of the adaptive filter input signal, one might consider the possibility of devising algorithms more closely related to the LMS algorithms than the ones mentioned in the previous paragraph. More specifically, suppose we have some a priori knowledge of input autocorrelation properties manifest in a collection of example autocorrelation matrices \{R_0, R_1, \ldots, R_{K-1}\}. Our goal is now to design an LMS-like algorithm whose convergence speed is optimized with respect to the above type of information about the autocorrelation properties of the input signal. It is hypothesized that if the autocorrelation properties of an actual input signal matches, or is close to, the ones specified through \{R_0, R_1, \ldots, R_{K-1}\}, then good convergence behavior is to be expected.

In the next section we develop the PLMS algorithms and show that the additional computational cost relative to that of the LMS algorithm is, in many cases, negligible. This is followed by some simulation results, before the paper is summarized and concluded.

2. THE PLMS ALGORITHM

In Fig.1 we show the prototypical adaptive filtering setup. As in all adaptive filtering applications our objective is to devise an online algorithm adjusting the coefficients of an FIR filter h(n), given as a length M column vector of filter coefficients, in such a way that the output signal, y(n), is a good estimate of the desired signal, d(n).
Motivation

We recall [1] that the LMS algorithm can be stated as
\[ h(n + 1) = h(n) + \mu \varepsilon(n)e(n), \]
where
\[ \varepsilon(n) = x(n) - h(n)^T x(n) \]
and the output error signal, \( e(n) \) is given by
\[ e(n) = d(n) - h^T(n)\varepsilon(n). \]
Defining the coefficient deviation as \( \varepsilon(n) = h(n) - h_t \), where \( h_t \) is the unknown true filter that we are trying to identify, it is not difficult to show that under the independence assumption [1]
\[ E\{\varepsilon(n + 1)\} = (I - \mu R)E\{\varepsilon(n)\}, \]
where \( E\{\} \) is the expectation operator, \( I \) denotes the identity matrix and \( R \) is the autocorrelation matrix defined by \( R = E\{\varepsilon(n)\varepsilon^T(n)\} \). The LMS/Newton algorithm which can be stated as [2]
\[ h(n + 1) = h(n) + \mu R^{-1}\varepsilon(n)e(n), \]
is the most efficient of all adaptive algorithms, see p. 11 of [6]. Since perfect knowledge of \( R \) and its inverse is not available, this algorithm has primarily theoretical interest as a yardstick to which other practical algorithms can be compared. Motivated by the LMS/Newton algorithm we propose an update equation which we will refer to as the PLMS algorithm:
\[ h(n + 1) = h(n) + \mu C \varepsilon(n)e(n), \]
where the matrix \( C \) is called a preconditioner [7]. We note that selecting \( C \) as \( I \), we get the LMS algorithm, whereas selecting \( C \) as the inverse of the autocorrelation matrix, had that been exactly known, results in the LMS/Newton algorithm [2]. From this it is evident that the LMS algorithm can be considered optimal if the input autocorrelation matrix is diagonal with all its diagonal elements being equal. If we have some a priori information about the autocorrelation structure of the input to the adaptive filter, one might wonder if it is possible to design some \( C \) matrix that, when employed in the update of Eq. 6, can lead to improved performance without significantly increasing computational complexity relative to the LMS algorithm? The answer is affirmative if the actual input signal has autocorrelation properties reasonably close to the autocorrelation properties assumed and the \( C \) matrix is constructed as a structured and/or sparse matrix approximating the inverses of assumed input autocorrelation matrices. Actually, our experiments indicate that the requirements on the degree of closeness need not be very strict.

In this paper, we assume that a priori knowledge of the autocorrelation properties are available through a collection of autocorrelation matrices \( \{R_0, R_1, \ldots, R_{K-1}\} \) which we consider as good approximations to autocorrelation matrices of inputs encountered in practice.

It is well known that the high speed of convergence of the LMS algorithm is dependent on the degree to which the eigenvalues of \( I - \mu R \) are clustered around zero. For the PLMS algorithms introduced here the convergence speed is clearly determined by the eigenvalue spread of \( I - \mu CR \). For the LMS/Newton algorithm clustering of the eigenvalues around zero is clearly achieved since \( C \) is selected as \( R^{-1} \). This is also what we attempt to do in the Recursive Least Squares (RLS) algorithm, where we maintain a time-variant estimate of the inverse autocorrelation matrix. As is well known, RLS is troubled by high computational complexity and numerical stability problems. Thus, we propose the use of a fixed \( C \) matrix that is a reasonably good approximate inverse to all typical autocorrelation matrices, \( \{R_0, R_1, \ldots, R_{K-1}\} \). In addition, to keep computational complexity low, we restrict \( C \) to have a simple structure specified through at most \( P \) scalar parameters given by the vector
\[ p = [p_0, p_1, \ldots, p_{(p-1)}]^T. \]
Identifying explicitly the rows of \( C \)
\[ C = \begin{bmatrix} \varepsilon_0^T \\ \varepsilon_1^T \\ \vdots \\ \varepsilon_{M-1}^T \end{bmatrix} \]
they can now be formed from \( p \) through
\[ \varepsilon_i = S_i p, i = 0, 1, \ldots, M - 1, \]
where \( S_i \) denote a binary 0/1 matrix with at most one non-zero entry in each row. We see that the function of these 0/1 matrices is to populate the entries of \( C \) with parameters from \( p \).

Clearly, a general matrix \( C \) will increase the computational complexity of the adaptive filter substantially. However, restricting \( C \) to be highly structured, or even better, highly structured and sparse will, when taking the shift structure of \( \varepsilon(n) \) into consideration, lead to modest or negligible complexity.
increases. For example, specifying \( C \) as circulant symmetric gives a multiplicative PLMS complexity of \( 2.5M \) compared to the \( 2M \) complexity of the LMS algorithm, whereas the structure for \( C \) to be presented below leads to a negligible complexity increase. Given this, our version of the PLMS algorithm has a computational complexity that, in practical terms, is the same as that of the ordinary LMS algorithm.

Having presented the basic idea underpinning PLMS, we briefly comment on its stability range before presenting a design procedure for \( C \) when its structure is given. This is followed by an example outlining the design of \( C \) when its structure is postulated to have the same structure as the structure of the inverse of the autocorrelation matrix of a first order autoregressive (AR(1)) signal. Note however, that the design procedure for \( C \) is usable for any selected structure.

**Selection of adaption step size, \( \mu \)**

In standard textbooks on adaptive filters, such as [1, 2], a stability condition for the LMS algorithm is found from Eq. 4 resulting in

\[
0 < \mu_{LMS} < \frac{2}{\lambda_{\text{max}}(R)},
\]

where \( \lambda_{\text{max}}(R) \) denotes the largest eigenvalue of matrix \( R \). In a similar manner, we derive for the PLMS algorithm the stability condition

\[
0 < \mu_{PLMS} < \frac{2}{\lambda_{\text{max}}(CR)}.
\]

This leads to somewhat different considerations in the determination of the adaption step size \( \mu \) for LMS and PLMS. Suppose we desire an adaptive algorithm with LMS-like complexity, but we know that the input signal will be AR(1) with correlation coefficients, \( \rho \), in the range 0.6 < \( \rho \) < 0.99. Using an LMS algorithm in such a situation will dictate a small \( \mu \) and slow convergence. Designing a \( C \) matrix for an AR(1) signal with, say, \( \rho = 0.85 \), would give optimum performance for inputs conforming to this hypothesis as well as good performance for inputs whose characteristics do not deviate too much from this assumption. Thus, unless \( C \) is not a really terrible approximation to the inverse of the autocorrelation matrix of expected input signals, we can expect \( \lambda_{\text{max}}(CR) < \lambda_{\text{max}}(R) \), making it possible to use larger adaption step sizes for PLMS than for LMS while maintaining comparable stability margins over the range of expected input signal autocorrelation properties.

**Design procedure for preconitioner \( C \)**

Based on the above, our problem can be summarized as follows: Given an assumed set of autocorrelation matrices for the input signal, i.e. given \( \{R_0, R_1, \ldots, R_{K-1}\} \), and a structured matrix \( C \) specified through the \( P \) free parameters of the \( p \) vector and the postulated/selected set of 0/1 structure determining matrices \( S_i, i = 0, 1, \ldots, M - 1 \), find the \( p \) that gives \( C \) as the best approximate inverse to the set of assumed autocorrelation matrices, i.e. solves the optimization problem

\[
\min_P \sum_{k=0}^{K-1} ||CR_k - I||_F^2, \text{ subject to } C = S_i p_i = 0, 1, \ldots, M - 1, \tag{10}
\]

where \( || \cdot ||_F \) denotes the Frobenius norm.

The objective function of Eq. 10 can be expressed in terms of the \( \xi_i \)-vectors constituting \( C \) as follows:

\[
\sum_{k=0}^{K-1} ||CR_k - I||_F^2 = \sum_{k=0}^{K-1} ||R_k C^T - I||_F^2 = \sum_{k=0}^{K-1} \sum_{i=0}^{M-1} ||R_k \xi_i - S_i p_i \xi_i||^2, \tag{11}
\]

where \( \xi_i \) is the length \( M \) column vector with a 1 in position \( i \) and zeros in all other positions. Making use of Eq. 9, the objective function can be written as

\[
\sum_{k=0}^{K-1} \sum_{i=0}^{M-1} ||R_k S_i p_i - \xi_i||^2.
\]

Minimizing this objective function corresponds to finding the least squares solution to the over determined equation set

\[
\begin{bmatrix}
R_0 S_0 \\
R_0 S_1 \\
\vdots \\
R_0 S_{M-1} \\
R_1 S_0 \\
\vdots \\
R_1 S_{M-1} \\
\vdots \\
R_{K-1} S_0 \\
\vdots \\
R_{K-1} S_{M-1}
\end{bmatrix}
\cdot
\begin{bmatrix}
\xi_0 \\
\xi_1 \\
\vdots \\
\xi_{M-1}
\end{bmatrix}
= \begin{bmatrix}
p \\
\vdots \\
\xi_{M-1}
\end{bmatrix}, \tag{12}
\]

where the least squares solution is found by solving the normal equations

\[
\begin{bmatrix}
R_0 S_0 \\
R_0 S_1 \\
\vdots \\
R_0 S_{M-1} \\
R_1 S_0 \\
\vdots \\
R_1 S_{M-1} \\
\vdots \\
R_{K-1} S_0 \\
\vdots \\
R_{K-1} S_{M-1}
\end{bmatrix}
\cdot
\begin{bmatrix}
\xi_0 \\
\xi_1 \\
\vdots \\
\xi_{M-1}
\end{bmatrix}
= \begin{bmatrix}
p \\
\vdots \\
\xi_{M-1}
\end{bmatrix}.
of such a signal is known to have form (we set

\[ s \in \mathbb{R}^N \text{ sive.} \]

Since the inverse of the autocorrelation matrix

\[ R^{-1} \]

As a specific example, let us assume that typical in-

\[ \sum_{k=0}^{M-1} R_k S_0 = \sum_{k=0}^{M-1} R_k S_1 = \ldots = \sum_{k=0}^{M-1} R_k S_{M-1} \]

Example

Specifying \( S_i \) matrices for \( C \) matrices having for

\[ a c b c 0 0 0 0 0 \]

\[ c b a c b c 0 0 0 \]

\[ 0 0 0 c b a c b c \]

\[ 0 0 0 0 0 c b c \]

\[ 0 0 0 0 0 0 0 c a \]

in this example):

\[ \begin{bmatrix} a & c & 0 & 0 & 0 & 0 & 0 \\ c & b & 0 & 0 & 0 & 0 & 0 \\ 0 & c & b & 0 & 0 & 0 & 0 \\ 0 & 0 & c & b & c & 0 & 0 \\ 0 & 0 & 0 & 0 & c & b & c \\ 0 & 0 & 0 & 0 & 0 & c & b \\ 0 & 0 & 0 & 0 & 0 & 0 & c \end{bmatrix} \]

in this example)

\[ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \]

and so on.

Formalizing this somewhat, it is readily seen that

\[ P = [a, b, c]^T \]

we can build \( C \) of the specified

\[ \begin{bmatrix} a \\ b \\ c \end{bmatrix} \]

with \( p \) = [a, b, c]^T, we can build C of the specified structure through the use of Eq. 9 with the following selections for \( S_0, S_1, \ldots, S_{M-1} \):

\[ S_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \]

\[ S_7 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \]

and

\[ S_i = Z^{i-1} S_1, i = 1, 2, \ldots, 6, \]

where

\[ S_1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \]
and

\[
Z = \begin{bmatrix}
0 & 0 & 0 & 0 & \cdots & 0 \\
1 & 0 & 0 & 0 & \cdots & 0 \\
0 & 1 & \ddots & \ddots & \ddots & \ddots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & 0 & \cdots & 1 & 0 & 0
\end{bmatrix}.
\] (18)

Before presenting experimental results indicating the performance of the PLMS algorithm, for the particular choice of structure of C made in the example above, we point out that for white input, i.e., \( R = I \) and \( K = 1 \), Eq. 13 implies that the optimum C matrix is the identity. This is hardly surprising, and we can say that the LMS algorithm is the optimum adaptive filter algorithm conforming to the structure of Eq. 6 when the input signal is white. If we know that the input signal will rarely be white, but rather have, or be close to, some other approximately known correlation structures given by some set of autocorrelation matrices \( \{R_0, R_1, \ldots, R_{K-1}\} \), it does not make sense to use the LMS algorithm. The theory presented above provides the tools necessary for the design of an appropriate adaptive algorithm in such situations.

3. EXPERIMENTAL RESULTS

As an example of the performance of PLMS we used the algorithm in system identification experiments as detailed below.

Optimized C for single R

In designing the C, a single assumed autocorrelation matrix R for an AR(1) signal with \( \rho = 0.85 \) was used. With this R matrix and the \( S_i \)-matrices specified in the example of the previous section, the solution for \( p \) in Eq. 13 was found. The designed C matrix was used in the PLMS algorithm applied to AR(1) inputs generated according to \( x(n) = \rho x(n-1) + w(n) \) with \( \rho \in \{0.75, 0.85, 0.95\} \) and where \( w(n) \) is a white, zero mean Gaussian signal. The signal is normalized, i.e., \( \sigma_x^2 = 1.0 \). Thus, we have inputs with various degrees of approximate match between the assumed autocorrelation used in the design of C and the autocorrelation of the input signals to which the PLMS algorithm is applied. In generating \( d(n) \), the true, but unknown filter vector \( h_c \), has \( M = 8 \) taps\(^1\) and is selected at random. The desired signal \( d(n) \) is thus given by \( d(n) = h_c^T x(n) + v(n) \), where \( v(n) \) is a white, zero mean Gaussian signal with \( \sigma_v^2 = 10^{-3} \). The learning curves, obtained in each case by averaging \( e^2(n) \) over 300 independent realizations, are shown in Fig. 2. Note that for both the LMS and PLMS curves we have used a value for \( \mu \) that is selected conservatively, \( \mu_{LMS} = 0.015 \) and \( \mu_{PLMS} = 0.03 \), in such a way that stability is also ensured for inputs with correlation properties very different from the ones used in producing the plots of Figs. 2. In the selection of \( \mu_{PLMS} \) and \( \mu_{LMS} \) we have taken care to ensure stability ranges of comparable extents for the two algorithms. This process involves an element of trial and error. From the figures it is evident that the PLMS algorithm exhibits excellent convergence behavior when we have reasonable match between assumed inputs and actually encountered inputs. The performance improvement relative to LMS is also evident, particularly for highly colored input signals.

Optimized C for multiple R matrices

We also performed an experiment in which we considered three autocorrelation matrices to be representative of the expected input autocorrelations. We employed autocorrelation matrices for AR(1) signals with \( \rho \in \{0.75, 0.85, 0.95\} \) in designing a single C matrix. This C matrix was used in generating learning curves for input AR(1) signals with \( \rho \in \{0.75, 0.85, 0.95\} \) which are shown in part a of Fig. 3. We notice rapid convergence in all cases, once again a significant improvement from ordinary LMS. Compared to the results of the previous subsection, we do not note any significant differences. This indicates that designing the C matrix for a single autocorrelation matrix and using it for inputs whose autocorrelation properties lie in the vicinity of the one for which C is optimized gives about the same improvement in convergence speed as when C is optimized with respect to a collection of closely related autocorrelation matrices.

Using a more diverse collection of assumed autocorrelation matrices, again for AR(1) signals now with \( \rho \in \{0.05, 0.1, 0.15, \ldots, 0.9, 0.95\} \) in the design of C, and solving for \( p \) using Eq. 13, we get learning curves as shown in part b of Fig. 3 when the inputs are AR(1) signals with \( \rho \in \{0.15, 0.75, 0.85\} \) when \( \mu_{PLMS} = 0.03 \). Comparing this with LMS learning curves obtained with \( \mu_{LMS} = 0.015 \) in order to maintain algorithm stability over the whole range of possible input signals – as shown in part c of Fig. 3 we see that PLMS obtains a more “equalized” convergence behavior throughout the whole region for which it is optimized than what is the case for the LMS. This comes at a negligible cost in computational complexity.

4. SUMMARY AND CONCLUSION

We have presented a new adaptive algorithm, PLMS, based on the use of a C matrix in a generic adaptive filter, Eq. 6. We have presented a design strategy, based on knowledge or approximate knowledge of possible input autocorrelation matrices, for the C matrix in which various structures and degrees

\(^1\)Other filter lengths were also simulated leading to the same type observations as presented here for \( M = 8 \).
of sparsity – implying low computational cost – are easily incorporated. The potential of the PLMS algorithm has been demonstrated through experiments in which we imposed the structure of the inverse of the autocorrelation matrix of an AR(1) signal on the C matrix. Future research includes the systematic exploration of different C matrix structures and their relative merits. Also, systematic and more formal guidelines for the selection of the adaption step size is under investigation.

We finally stress what we consider a fundamental issue: We have presented an optimization problem for the C matrix in the generic recursion given by Eq. 6. When the assumed autocorrelation matrix for the input signal is diagonal with equal entries on the diagonal, i.e. we have white inputs, the LMS algorithm has optimum performance in terms of speed of convergence. On the other hand, for other assumptions on the input autocorrelation matrix, we have pointed out that improved performance can be obtained within the PLMS framework. If such knowledge of the autocorrelation properties of the input signal can be assumed in a given application, it should be exploited. In practical terms the question to be asked when considering a simple adaptive algorithms for some practical application is not: “Will the LMS algorithm do the job?”, but rather “Can assumptions about the input autocorrelations be made such that a PLMS algorithm with an appropriately selected C matrix will do the job?”

Fig. 2: Learning curves for the LMS algorithm and the PLMS algorithm with C matrix designed for an assumed input autocorrelation matrix of an AR(1) signal with $\rho = 0.85$. The input signals were selected as AR(1) signals with a) $\rho = 0.75$, b) $\rho = 0.85$, and c) $\rho = 0.95$. In all cases the PLMS algorithm gives the lower curves. For both the LMS and PLMS plots we have selected $\mu$ conservatively, $\mu_{\text{LMS}} = 0.015$ and $\mu_{\text{PLMS}} = 0.03$. That is, we have ensured a reasonable, and for the two algorithms comparable, margin to prevent instability for a wide range of input autocorrelation properties.

Fig. 3
A Preconditioned LMS adaptive filter

(b) Learning Curve

AR(1) input with \( p = 0.15 \) and \( p = 0.75 \)

AR(1) input with \( p = 0.85 \)

(c) Learning Curve

AR(1) input with \( p = 0.15 \)

AR(1) input with \( p = 0.75 \)

AR(1) input with \( p = 0.85 \)

Fig. 3: a) Learning curves for the PLMS with \( C \) matrix designed for three assumed input autocorrelations of AR(1) signal with \( \rho \in \{0.75, 0.85, 0.95\} \). The input signals were selected as AR(1) processes with \( \rho = 0.75, \rho = 0.85, \) and \( \rho = 0.95 \). We note considerable improvement in convergence speed relative to the LMS, – see Fig. 2. b) Learning curves for the PLMS algorithm with \( C \) matrix designed for an assumed collection of input autocorrelation matrices of an AR(1) signal with \( \rho \in \{0.05, 0.1, 0.15, \ldots, 0.9, 0.95\} \). The input signals for this PLMS algorithm were selected as AR(1) signals with \( \rho \in \{0.15, 0.75, 0.85\} \). c) Learning curves for LMS with same inputs as described in b) above. As explained in the text, \( \mu_{PLMS} = 0.03 \) and \( \mu_{LMS} = 0.015 \) in all cases.

References


John Håkon Husøy received the M.Sc. and Ph.D. degrees in electrical engineering from the Norwegian University of Science and Technology. In his early career he was involved in hardware and software development in various positions in several companies in Canada and Norway. Since 1992 he has been a Professor with the Department of Electrical and Computer Engineering, University of Stavanger, Norway. His research interests include adaptive algorithms, digital filtering, signal representations, image compression, bioelectrical signal processing, and image analysis.