Channel Equalization Using a New Transform Domain LMS Algorithm with Adaptive Step-Size

RADU CIPRIAN BILCU, PAULI KUOSMANEN, KAREN EGIAZARIAN
1,2,3Signal Processing Laboratory,
Tampere University of Technology
P.O.BOX 553, FIN-33101, Tampere,
FINLAND

Abstract: - In a recent paper the authors proposed a new Transform Domain LMS algorithm with variable step-size (TDVSS). As a novelty, comparing with the existing Transform Domain LMS algorithms, the new TDVSS uses an adaptive step-size that depends also on the output error. In this paper, we study the theoretical behavior and implementation of TDVSS for the problem of channel equalization. The speed of convergence and the steady-state misadjustment are compared with that of the most known Transform Domain and Time Domain LMS algorithms.

Key-Words: - Transform Domain LMS, Variable Step-Size LMS, Transform Domain Variable Step-Size LMS algorithm, Channel Equalization.

1 Introduction

In the case of communication channels the transmitted data can be distorted due to the channel noise and also due to the time dispersion of the channel response. The channel transfer function produces the so-called intersymbol interference (ISI), that limits the maximum transmission speed of data. One way to eliminate the ISI is to perform an adaptive channel equalization that uses at the receiver a filter whose coefficients are adapted, such that the ISI is eliminated or reduced.

The conventional LMS adaptive algorithm [1] has the advantage of being simple to implement. However, the LMS algorithm with fixed step-size [1] have the disadvantage that for a small value of the step-size, its convergence speed is small. If the step-size is increased in order to increase the convergence speed, then the adaptation error will also increase. In order to deal with this problem many adaptive filters with variable step-sizes were proposed in the literature [1]-[3].

Another drawback of the LMS and its time domain implementations is that their convergence speed depends on the eigenvalue spread of the input autocorrelation matrix. If the eigenvalue spread is large, then the LMS filter has a very slow convergence. One way to eliminate this problem is to prewhiten the input signal using a unitary transform, followed by the normalization by the power of the transform coefficients. Adaptive filters obtained this way form the class of Transform Domain LMS (TDLMS) (see [4]-[6] and the references therein). In a recent paper [7], the authors proposed a new Transform Domain LMS algorithm - the Transform Domain Variable Step LMS algorithm (TDVSS). The main difference between the new TDVSS algorithm and the existing transform domain implementations is that its step-size depends also on the output error.

In this paper we study the behavior of the new algorithm in the case of channel equalization problem. The simulations presented here shows that the new algorithm has better convergence speed than the other well known adaptive filters.

2 Channel Equalization Using TDVSS

The block diagram for channel equalization using the Transform Domain LMS algorithm is depicted in Fig. 1, where \( x(n) \) is the train signal applied at the input of the transmission channel, \( v(n) \) is the
channel noise, the block denoted by \( T \) represents the transform layer applied to the input signal, \( s(n) \) represents the vector of the transform coefficients, \( y(n) \) is the output of the adaptive filter, \( D \) is the number of delays applied to the training signal and \( d(n) = x(n - D) \) is the desired signal.

In the standard Transform Domain LMS algorithm, the coefficients are updated as follows:

\[
\hat{h}(n + 1) = \hat{h}(n) + \mu \Gamma^{-1}(n)s(n)e(n) \tag{1}
\]

where \( \hat{h}(n) = [\hat{h}_1(n), \hat{h}_2(n), \ldots, \hat{h}_N(n)]^T \) is the \( N \times 1 \) vector containing the filter coefficients, \( e(n) \) is the output error, \( \Gamma(n) \) is a diagonal matrix having on the main diagonal the power estimates of the transform coefficients \( s_i(n) \) and \( \mu \) is the step-size.

Usually, for estimation of the powers of \( s_i(n) \), the following recursive formula is used:

\[
\Gamma_i(n + 1) = \alpha \Gamma_i(n) + (1 - \alpha) s_i(n)^2, \quad i = 1, N, \tag{2}
\]

where \( \alpha \in (0, 1) \) is a constant parameter and \( \Gamma_i(n) \) is the \( i^{th} \) diagonal element of \( \Gamma(n) \).

Analyzing (1) we can see that each coefficient is updated using a distinct step-size given by:

\[
\mu_i(n) = \frac{\mu}{\Gamma_i(n)}, \quad i = 1, N. \tag{3}
\]

The step-sizes \( \mu_i(n) \) defined by (3) depend on the value of \( \Gamma_i(n) \) therefore, one can consider that \( \mu_i(n) \) is also time-variable. For stationary input signals it is well known that after a number of iterations the power estimates stabilizes to some values near the true powers therefore, the step-sizes \( \mu_i(n) \) become constants (they have small oscillations around some constant values). It was proved in the literature, that the use of an adaptive step-size for the LMS algorithm can improve its steady-state misadjustment and also its convergence rate. Based on this observations, in [7] a new Transform Domain LMS algorithm was introduced. In the new TDVSS algorithm each step-size \( \mu_i(n) \) depends not only on the corresponding power estimate but also on the output error. In [7] the behavior of the Mean Square Error for the case of system identification was studied. The simulations showed that the new algorithm performs better than the most known time domain and transform domain implementations of the LMS adaptive filter.

In this paper the theoretical behavior of the TDVSS algorithm is derived. The new algorithm is used also for the problem of channel equalization and it shows again better performance than the other well known adaptive filters.

The new TDVSS filter uses the following formula for update the coefficients:

\[
\hat{h}(n + 1) = \hat{h}(n) + \mu(n)\Gamma^{-1}(n)s(n)e(n) \tag{4}
\]

whith \( \mu(n) \) given by:

\[
A(n) = \beta \mu(n) - 1 + \frac{\gamma}{L} \sum_{i=n-L}^{n-1} e^2(i)
\]

\[
\mu(n) = \begin{cases} 
\mu_{min}, & \text{if } n - 1 = L, 2L, \ldots \text{ and } A(n) \in [\mu_{min}, \mu_{max}] \\
\mu_{min}, & \text{if } n - 1 = L, 2L, \ldots \text{ and } A(n) < \mu_{min} \\
\mu_{max}, & \text{if } n - 1 = L, 2L, \ldots \text{ and } A(n) > \mu_{max} \\
\mu(n-1), & \text{otherwise}
\end{cases} \tag{5}
\]

where \( \beta \in (0, 1), \gamma \in (0, 1) \) being some constant parameters and \( L \) is an integer parameter equal with the number of samples in which the average of the square error is computed.

The equations describing the TDVSS are the same as for the standard TDLMS algorithm. The only difference is that the constant component \( \mu \) of
the step-size in (1) is now adaptive and it is modified using (5). Thus, the coefficients of the new algorithm are computed using (4). We note that the value of \( \mu(n) \) in (4) is not changed at each iteration but is constant for \( L \) consecutive iterations and after that it is updated by (5) using the output error.

### 3 The Steady-State Mean Square Error Analysis for the New TDVSS Algorithm

Equation (4) can be rewritten as follows:

\[
\hat{h}(n+1) = \hat{h}(n) + \bar{\mu}(n)s(n)e(n)
\]

where \( \bar{\mu}(n) = \mu(n)\Gamma^{-1}(n) \) is the \( N \times N \) diagonal matrix with the diagonal elements given by \( \bar{\mu}_i(n) = \mu_i(n)\Gamma^{-1}_i(n) \) and \( \mu(n) \) is given in (5).

To make the convergence analysis of the TDVSS algorithm more tractable, besides the usual assumptions, we introduce the following ones:

**Assumption 1:** After few iterations the power estimates of the transform coefficients \( s_i(n) \) become constants and, therefore, the step-sizes \( \bar{\mu}_i(n) \) are independents from \( s(n) \).

**Assumption 2:** The step-sizes \( \bar{\mu}_i(n) \) in (6) and the output error \( e(n) \) are independent. This can be justified by the fact that the update of the step-size is done using just some past \( L \) values of the error (see (5)).

The Mean Square Error (MSE) is defined by:

\[
J(n) = E \left\{ e^2(n) \right\} = E \left\{ (d(n) - \bar{y}(n))^2 \right\},
\]

\[
= E \left\{ (d(n) - y_0(n) + y_0(n) - \bar{y}(n))^2 \right\},
\]

\[
= E \left\{ (e_0(n) - \Delta y(n))^2 \right\},
\]

where \( y_0(n) \) is the optimum output obtained when the coefficients of the adaptive filter are equals with the optimum Wiener solution \( h_0 \) and \( e_0(n) = d(n) - y_0(n) \) is the error in the case of optimum adaptation. The quantity \( \Delta y(n) \) can be written as:

\[
\Delta y(n) = y(n) - y_0(n) = \hat{h}^t(n)s(n) - h_0^t s(n)
\]

\[
\Delta y(n) = (\hat{h}(n) - h_0)^t s(n) = \Delta h^t(n)s(n)
\]

Substituting (8) to (7) and using the fact that the error \( e_0(n) \) is orthogonal to the transform coefficients \( s_i(n) \) and independent of \( \Delta h_i(n) \), we can write:

\[
J(n) = E \left\{ e^2_0(n) \right\} + E \left\{ \Delta y(n)^2 \right\},
\]

\[
= J_{\text{min}} + E \left\{ \Delta h^t(n)s(n)s^t(n)\Delta h(n) \right\},
\]

\[
= J_{\text{min}} + \text{tr} \left[ R_{ss} C(n) \right],
\]

where \( C(n) = E \left\{ \Delta h \Delta h^t \right\} \) is the covariance matrix of the weight error vector, \( R_{ss} \) is the autocorrelation matrix of the transform coefficients vector defined by \( E \left\{ s(n)s^t(n) \right\} \), \( \text{tr} \left[ \cdot \right] \) denotes the trace operator of a matrix and \( J_{\text{min}} = E \left\{ e^2_0(n) \right\} \) is the minimum mean square error.

The autocorrelation matrix can be expressed as \( R_{ss} = QA_{ss}Q^t \), where \( A_{ss} = \text{diag} \left[ \lambda_0, \ldots, \lambda_{N-1} \right] \) is the diagonal matrix having on the main diagonal the eigenvalues of \( R_{ss} \). \( Q \) is the modal matrix of \( R_{ss} \), \( QQ^t = I \) and \( Q^{-1} = Q^t \). Denoting \( C'(n) = QC(n)Q^t \), (9) can be rewritten as follows:

\[
E \left\{ e^2(n) \right\} = J_{\text{min}} + \text{tr} \left[ A_{ss} C'(n) \right]
\]

\[
E \left\{ e^2(n) \right\} = J_{\text{min}} + \sum_{i=0}^{N-1} \lambda_i c_{ii}'(n)
\]

where \( c_{ii}'(n) \) are the diagonal elements of \( C'(n) \).

The weight error vector is given by:

\[
\Delta h(n+1) = \hat{h}(n) - h_0 + \bar{\mu}(n)s(n)e(n)
\]

\[
\Delta h(n+1) = \left[ I - \bar{\mu}(n)s(n)s^t(n) \right] \Delta h(n) + \bar{\mu}(n)e_0(n)s(n)
\]

Computing the outer product of (11) by itself, taking the expectations on both sides and using \( C'(n) = QC(n)Q^t \) one obtains:

\[
C'(n+1) = C'(n) - E \left\{ \bar{\mu}(n) \right\} \left[ C'(n)A_{ss} + A_{ss}C'(n) \right] + E \left\{ \bar{\mu}^2(n) \right\} J_{\text{min}} A_{ss} + 2A_{ss}C'(n)A_{ss} + \text{tr} \left[ A_{ss} C'(n) A_{ss} \right]
\]

Thus diagonal elements \( c_{ii}'(n) \) of the matrix \( C'(n) \) are given by:

\[
c_{ii}'(n+1) = [1 - 2E \left\{ \bar{\mu}_i(n) \right\} \lambda_i] c_{ii}'(n) + 2E \left\{ \bar{\mu}_i^2(n) \right\} \lambda_i^2 c_{ii}'(n) + E \left\{ \bar{\mu}_i^2(n) \right\} J_{\text{min}} + \sum_{m=0}^{N-1} E \left\{ \bar{\mu}_m^2(n) \right\} \lambda_m c_{mm}'(n)
\]
Following the same derivations as in [3] and [6] the sufficient condition that ensures convergence of the mean squared error is:

\[ 0 < E \left[ \frac{\mu^2(\infty)}{E[\mu(\infty)]]} \right] < \frac{2}{3tr[R_{ss}]} \]  

(14)

where \( E[\mu(\infty)] \) and \( E[\mu^2(\infty)] \) are the steady-state values of \( E[\mu(n)] \) and \( E[\mu^2(n)] \).

The mean value of the variable step-size \( \mu_i(n) \) is given by:

\[ E \{ \mu_i(n) \} = E \{ \mu(n) \Gamma_i^{-1}(n) \} = \frac{E \{ \mu(n) \}}{E \{ \Gamma_i(n) \}}, \]  

(15)

with \( \mu(n) \) given by:

\[ E \{ \mu(n) \} = \begin{cases} 
E \{ A(n) \}, & \text{if } \begin{cases} n - 1 = L, 2L, \dots \text{ and } A(n) \in [\mu_{min}, \mu_{max}] \\
\mu_{min}, & \text{if } \begin{cases} n - 1 = L, 2L, \dots \text{ and } A(n) < \mu_{min} \\
\mu_{max}, & \text{if } \begin{cases} n - 1 = L, 2L, \dots \text{ and } A(n) > \mu_{max} \\
E \{ \mu(n-1) \}, & \text{otherwise} 
\end{cases} 
\end{cases} 
\end{cases} \]  

(16)

and:

\[ E \{ A(n) \} = \beta E \{ \mu(n-1) \} + \]  

\[ + \frac{\gamma}{L} \sum_{k=n-L}^{n-1} e^2(k) \]

\[ = \beta \frac{E \{ \mu(n-1) \} + \gamma}{L} \sum_{k=n-L}^{n-1} J(k) \quad \ldots (20) \]

The mean square value of the step-size \( \mu_i(n) \) is given by:

\[ E \left\{ (\mu_i(n))^2 \right\} = \frac{E \{ \mu^2(n) \}}{E \{ \Gamma_i^2(n) \}} \]  

(18)

where

\[ E \{ \mu^2(n) \} = \begin{cases} 
E \{ A^2(n) \} & \text{if } \begin{cases} n - 1 = L, 2L, \dots \text{ and } A(n) \in [\mu_{min}, \mu_{max}] \\
\mu_{min}^2 & \text{if } \begin{cases} n - 1 = L, 2L, \dots \text{ and } A(n) < \mu_{min} \\
\mu_{max}^2 & \text{if } \begin{cases} n - 1 = L, 2L, \dots \text{ and } A(n) > \mu_{max} \\
E \{ \mu^2(n-1) \} & \text{otherwise} 
\end{cases} 
\end{cases} 
\end{cases} \]  

(19)

and:

\[ E \{ A^2(n) \} = \beta^2 E \{ \mu(n-1) \} + \]  

\[ + \frac{2\beta\gamma}{L} \sum_{k=n-L}^{n-1} e(k)^2 \]

\[ + \frac{\gamma^2}{L^2} \sum_{k=n-L}^{n-1} e^2(k) \]

(20)

Since \( L > \gamma \) and \( E \{ \sum_{k=n-L}^{n-1} e^2(k) \} \) have small values at the steady-state, the last term in (20) can be discarded. Using assumption 2 one obtains:

\[ E \{ A^2(\infty) \} = \beta^2 E \{ \mu^2(\infty) \} + \]  

\[ + \frac{2\beta\gamma}{L} E \{ \mu(\infty) \} J(\infty) \quad \ldots (21) \]

Finally, we are interested in the steady-state Mean Square Error that can be computed from (10), (13), (15)-(21) when \( n \to \infty \). Combining (15), (16) and (17), the mean value of \( \mu_i \) at steady-state become:

\[ E \{ \mu_i(\infty) \} = \frac{\gamma J(\infty)}{\Gamma_i(\infty) (1 - \beta)}. \]  

(22)

The mean square value of \( \mu_i \) at steady-state can be obtained from (18), (19) and (21) when \( n \to \infty \), and is given by:

\[ E \left\{ \mu^2_i(\infty) \right\} = \frac{2\beta\gamma^2 J^2(\infty)}{\Gamma_i^2(\infty) (1 - \beta^2) (1 - \beta)}. \]  

(23)
where \( \Gamma_i(\infty) \) is the steady-state value of the \( i^{th} \) diagonal element of \( \Gamma(n) \) and it is equal to the power of the \( i^{th} \) transform coefficient \( s_i(n) \). In (22) and (23) we have assumed that the step-size is in between \( \mu_{\min} \) and \( \mu_{\max} \). The steady state misadjustment can be obtained from (10) and (13) using the same derivations as in [3] and [6], and it is given by:

\[
M = \frac{J(\infty) - J_{\min}}{J_{\min}} \approx \frac{1}{2} \sum_{i=0}^{N-1} y_i \lambda_i,
\]

where \( y_i = \frac{E\{\hat{\mu}_i^2(\infty)\}}{E\{\hat{\mu}_i(\infty)\}} \), \( E\{\hat{\mu}_i(\infty)\} \) and \( E\{\hat{\mu}_i^2(\infty)\} \) are given by (22) and (23) respectively, and \( \lambda_i \) are the eigenvalues of the matrix \( R_{ss} \). After some simple mathematical manipulations, (24) becomes:

\[
M \approx \frac{\beta \gamma N}{1 - \beta^2} \frac{J_{\min}}{1 - \frac{\beta \gamma N}{1 - \beta^2} J_{\min}} \sum_{i=0}^{N-1} \lambda_i \Gamma_i(\infty)
\]

(25)

Since the matrix \( R_{ss} \) is near diagonal and the power estimates \( \Gamma_i(\infty) \) are close to the real powers, then the summation in (25) can be approximated with \( N \). Thus finally, the steady-state misadjustment can be written as follows:

\[
M \approx \frac{\beta \gamma N \mu_{\min}}{1 - \beta^2} \left( \frac{1}{2} \right)^{\frac{\beta \gamma N}{1 - \beta^2} \mu_{\min} N}
\]

(26)

which is similar to the results derived in [3]. We note that for a constant step-size, say \( \mu(\infty) = \mu_{\min} \), equation (24) becomes:

\[
M \approx \frac{1}{2} \mu_{\min} N
\]

(27)

that is the well known approximation for the misadjustment of the standard TDLMS.

The value of the parameter \( L \) in (5) is not critical for the algorithm. Actually we have seen in our simulations that \( L \) has to be smaller than the convergence time of the algorithm in order to have enough step-size updates. Any values \( L \leq 10 \) seems to be good choices for a wide range of applications.

Just for comparison purposes, in Table 1 the number of arithmetic operations required by the plain TDLMS, DCT-LMS from [6] and the new TDVSS algorithms at each iteration are presented. We can see from Table 1, that the new algorithm has a complexity less than DCT-LMS with \( M = 5 \) as proposed in [6] and is comparable with the complexity of the plain TDLMS.

### Table 1: Computational complexity of TDLMS, DCT-LMS and TDVSS

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>TDLMS</th>
<th>DCT-LMS</th>
<th>TDVSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mult./Div.</td>
<td>(6N + 1)</td>
<td>(5N + M_N)</td>
<td>(6N + 4)</td>
</tr>
<tr>
<td>Add./Sub.</td>
<td>(3N)</td>
<td>(3N + M_N)</td>
<td>(3N + 2)</td>
</tr>
</tbody>
</table>

Table 1: Computational complexity of TDLMS, DCT-LMS and TDVSS.

### 4 Simulations and Results

The new TDVSS algorithm was tested in channel equalization framework. The block diagram of the system used in our simulations is depicted in Fig. 1. The compared algorithms were: the plain LMS, the Variable Step-Size LMS proposed in [3], the correlation-based variable step LMS (MVSS) from [2], the plain TDLMS using the DCT transform, the DCT-LMS using the modified power estimator proposed in [6] and the new TDVSS.

The transform used in the simulations was the DCT. The signal to noise ratio at the output of the channel was \( S/N = 30\) dB. The test signal \( x(n) \) was binary with the samples randomly chosen from \{-1, +1\}. The parameters of all the compared algorithms were chosen such that they have comparable steady-state Mean Squared Errors.

The plotted learning curves were obtained by averaging the squared errors of 200 independent runs with each run containing a number of \( 15 \times 10^4 \) iterations. The steady-state mean squared errors are given in Table 2 and these values were obtained by averaging the last 1000 values from the corresponding MSEs.

### Table 2: The steady-state Mean Squared Errors

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>LMS</th>
<th>VSS</th>
<th>MVSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE</td>
<td>0.0155</td>
<td>0.0155</td>
<td>0.0157</td>
</tr>
<tr>
<td>Algorithm</td>
<td>TDLMS</td>
<td>DCT-LMS</td>
<td>TDVSS</td>
</tr>
<tr>
<td>MSE</td>
<td>0.0153</td>
<td>0.0161</td>
<td>0.0153</td>
</tr>
</tbody>
</table>

Table 2: The steady-state Mean Squared Errors

All the adaptive filters have the same number of coefficients \( N = 17 \). The transmission channel has three coefficients given by the following model (see [6]):

\[
h_i = \begin{cases} 
\frac{1}{2} \left[ 1 + \cos \left\{ \frac{2\pi}{N} (n - 2) \right\} \right], & \text{if } n = 1, 2, 3, \\
0, & \text{otherwise}
\end{cases}
\]

(28)

In Fig. 2 the learning curves obtained for the LMS, VSLMS, MVSS and TDVSS algorithms are
presented. In order to have a more clear representation, just the first 4000 samples of each learning curve are plotted. We can see from this figure that the TDVSS clearly has much better speed performances that the time domain implementations. This is expected since the input signal into the adaptive filter was highly correlated ($W = 3.75$ in (28)) and there is already proved in the literature that the transform domain implementations performs better in this cases (see e.g. [4]).

A more interesting result is presented in Fig. 3, were the TDVSS algorithm is compared with the plain TDLMS and the DCT-LMS using the modified power estimator. We can see from this figure that the TDVSS is the fastest algorithm among these three transform domain implementations.

5 Conclusions

In this paper the theoretical analysis of a new Transform Domain LMS adaptive filter with adaptive step-size is presented. Some guidelines for selecting the values of the parameters to obtain a desired steady-state MSE are also given. The new algorithm was applied to the problem of channel equalization and it shows much better convergence speed compared with other well known time domain and also transform domain algorithms. The computational complexity of the TDVSS algorithm is comparable with that of the plain TDLMS, which makes it a very good candidate for practical implementations.

References:


