Equalisation with adaptive time lag

This item was submitted to Loughborough University's Institutional Repository by the/an author.


Additional Information:

- (c) 2005 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works.

Metadata Record: https://dspace.lboro.ac.uk/2134/25677

Version: Accepted for publication

Publisher: © IEEE

Please cite the published version.
Equalization with adaptive time lag

Yu Gong and Colin F. N. Cowan

Institute of Electronics, Communications and Information Technology
The Queen’s University of Belfast
Northern Ireland Science Park
Queen’s Road, Queen’s Island Belfast BT3 9DT
United Kingdom
email: y.gong, c.f.n.cowan@ee.qub.ac.uk

Preprint for IEE Proceedings - Communications (final manuscript)

Abstract

In an adaptive equalizer, the time lag is an important parameter that significantly influences the performance. Only with the optimum time lag that corresponds to the best minimum-mean-square-error (MMSE) performance, can we have best use of the available resources. Many designs, however, choose the time lag either based on pre-assumption of the channel or simply based on average experience. In this paper, first we investigate the relation between the MMSE performance and the time lag using a new interpretation of the MMSE equalizer, and then propose a novel adaptive time lag algorithm based on gradient search. The proposed algorithm can converge to the optimum time lag in the mean and is verified by the numerical simulations provided in this paper.

Key words: Time lag, MMSE equalizer, adaptive algorithm
1 Introduction

Linear equalization is widely used in digital communications to combat intersymbol interference (ISI). Of particular interest is the adaptive MMSE equalizer due to its adaptability to a priori unknown channels. The MMSE equalizer usually has finite length, or finite impulse response (FIR), with coefficients updated by adaptive algorithms such as least mean square (LMS) or recursive least square (RLS) algorithms [1]. A traditional adaptive equalizer performs in two stages: at the first stage, pilot symbols are sent to train the equalizer; at the second stage, information bearing signals are transmitted and the equalizer operates on a decision-directed mode. Fig. 1 illustrates the adaptive equalizer corresponding to the training stage, where $x(n)$ is the information signal, $H(z)$ is the channel transfer function, $\eta(n)$ is the channel noise, $y(n)$ is the received signal, $\Delta$ is the time lag, $d(n) = x(n - \Delta)$ is the reference signal, $z(n)$ is the equalizer output, $e(n)$ is the error signal, and $W(z)$ is the equalizer.

To preserve the causality of the FIR equalizer, the time lag, or the decision delay, is introduced. The value of the time lag must be carefully chosen as it greatly influences the performance: First, there exists a minimum MMSE with respect to the time lag, and the MMSE for different time lag can vary significantly especially when the tap-length is short [2]. Second, when the tap-length is large, the equalizer may not be sensitive to some choices of the time lag, as was shown in [3] that adjusting the number of
taps in the decision feedback equalizer (DFE) can make it robust to variation of the decision delay. Under such a situation, plotting the MMSE versus the time lag will show a graph with a “flat line” around the time lag that minimizes the MMSE. Obviously, it is more desirable to set the time lag at the minimum value among those within the “flat line” rather than the time lag that minimizes the MMSE, by which we can have a minimum-delayed equalizer without sacrificing the MMSE performance. This is particularly important for a time critical system. For example, for a minimum-phase channel, the best time lag should be zero, though there may exist other choices of lags with similar, if not smaller, MMSE. In general, only if the time lag is set to the most “proper” value, can we have best use of the resources once the structure of the adaptive equalizer is fixed. We note that, though the time lag also affects the eigenvalue spread of the input correlation matrix, and then the convergence behavior, of the MMSE DFE [4], it has no effect on that of the linear MMSE equalizer whose input correlation matrix is only determined by the received signal vector.

Usually the value of the optimum time lag is between zero and the equalizer length, but the exact value depends greatly on the channel impulse response. In many applications, however, the time lag is chosen based on either the pre-assumption of the channel or simply from experience. For instance, in an experiment shown in [1, Section 9.7], the channel was assumed to be symmetric and the time lag was set equal to half of the tap-length plus the delay of the channel’s strongest pulse. This method, obviously, can not be generalized. In general, even with accurate channel estimation, it is still not straightforward to set an appropriate time lag. There are several publications related to this issue, for example, Lopez-Valcarce (et al.) showed in [5] that the optimum time lag for a blind constant modulus equalizer can be implied by choosing the best initialization
of the pre-whitened equalizer. In [6], Voois (et al.) expressed the time lag as an explicit parameter for the DFE and obtained the optimum delay after the performance for all possible delays were calculated and compared. In [7], Al-Dhahir (et al.) proposed an efficient method to calculated the optimum delay for the DFE. Some other articles, e.g. [8–10], investigated the time delay for multichannel equalizers. All these algorithms, however, are limited to specific applications and cannot be used by the linear MMSE equalizer. Moreover, these algorithms have high complexity, since either they are based on brute force searching or require manipulation of the input matrices, making them inefficient for on-line applications.

In this paper, we propose a novel adaptive time lag algorithm for the MMSE equalizer. The proposed algorithm is based on the observation that though the closed-form for the MMSE function of the time lag is difficult, if not impossible, to obtain, the relationship between the MMSE and time lag can be revealed instantaneously since the time lag is only a one dimensional parameter. Moreover, although the time lag must be an integer, we can apply the idea of the pseudo fractional time lag to make instantaneous adaption possible, where the true time lag is the integer part of the fractional lag.

The rest of this paper is organized as follows: Section 2 introduces a new interpretation of the MMSE equalizer; Section 3 investigates the relation between the MMSE and time lag; Section 4 proposes a novel adaptive time lag algorithm; Section 5 gives some numerical examples; Finally section 6 summarizes the paper. For clarity of exposition, the analyses in this paper are based on the real-valued signals, but results can be readily extended to complex signals.
2 System Model

This section introduces the basic structure of the adaptive MMSE equalizer, especially the unconstrained MMSE equalizer, and presents a new interpretation of the MMSE equalizer which is very useful for investigating the relation between the MMSE and time lag.

2.1 Basic Structure

The MMSE equalizer is obtained by minimizing the MSE cost function:

\[ \xi = E\{d(n) - w^T(n)y(n)\}^2 \]  

(1)

with respect to the tap-vector \( w(n) \), where \( y(n) = [y(n), \cdots, y(n - N + 1)]^T \) and \( N \) is the tap-length [1]. For later use, we introduce the unconstrained MMSE equalizer which has range extending from \(-\infty\) to \(\infty\). Specifically, assuming the time lag is \( \Delta \), the optimum unconstrained tap-vector can be expressed as:

\[ w_{\infty,\Delta} = [w_{\infty,\Delta}(-\infty), \cdots, w_{\infty,\Delta}(i), \cdots, w_{\infty,\Delta}(\infty)]^T, \]  

(2)

the transfer function of which is given by (see [11]):

\[ W_{\infty,\Delta}(z) = \frac{H^*(z)\Phi_{xx}(z) \cdot z^{-\Delta}}{\Phi_{xx}(z)|H(z)|^2 + \Phi_{\eta\eta}(z)} \]  

(3)
where $\Phi_{xx}(z)$ and $\Phi_{\eta\eta}(z)$ are $z$-transforms of $E[x^2(n)]$ and $E[\eta^2(n)]$ respectively. It can be easily obtained from (3) that:

$$w_{\infty,\Delta}(i) = w_{\infty,0}(i - \Delta),$$

$$\xi_{\infty,\Delta} = \xi_{\infty,0}$$

where $w_{\infty,0}(i)$ is the $i$th coefficient of the unconstrained equalizer with time lag at 0, $\xi_{\infty,\Delta}$ and $\xi_{\infty,0}$ are the MMSE for the unconstrained equalizers with time lags at $\Delta$ and 0 respectively.

2.2 New Interpretation

First, we define the following vectors as:

$$\tilde{y}(n) = [y_T^T(n) \ y^T(n) \ y_R^T(n)]^T,$$

$$w_{\infty,\Delta} = [w_{\infty,\Delta L}^T \ w_{\infty,\Delta N}^T \ w_{\infty,\Delta R}^T]^T,$$

where $y_L(n) = [y(\infty), \cdots, y(n + 1)]^T$, $y_R(n) = [y(n - N), \cdots, y(-\infty)]^T$, $w_{\infty,\Delta L} = [w_{\infty,\Delta}(-\infty), \cdots, w_{\infty,\Delta}(-1)]^T$, $w_{\infty,\Delta N} = [w_{\infty,\Delta}(0), \cdots, w_{\infty,\Delta}(N - 1)]^T$ and $w_{\infty,\Delta R} = [w_{\infty,\Delta}(N), \cdots, w_{\infty,\Delta}(\infty)]^T$. Then (1) can be expressed as:

$$\xi = E[e_{\infty,\Delta}(n) + w_{\infty,\Delta}^T \tilde{y}(n) - w^T(n)y(n)]^2,$$

where $e_{\infty,\Delta}(n) = d(n) - w_{\infty,\Delta}^T \cdot \tilde{y}(n)$. Since $e_{\infty,\Delta}(n)$ is the optimum error signal for the unconstrained equalizer, with the orthogonality principle, we have $E[e_{\infty,\Delta}(n) \cdot y(n)] = 0$
for all \( n \). Then (6) becomes:

\[
\xi = \xi_{\infty, \Delta} + \mathbb{E}|w_{\infty, \Delta}^T \tilde{y}(n) - w^T(n)y(n)|^2, \tag{7}
\]

where \( \xi_{\infty, \Delta} = \mathbb{E}|e_{\infty, \Delta}(n)|^2 \). But from (4) that \( \xi_{\infty} = \xi_{\infty, \Delta} \), thus minimizing (7) or:

\[
\tilde{\xi} = \mathbb{E}|\tilde{d}(n) - w^T(n)y(n)|^2 \tag{8}
\]

with respect to \( w(n) \) gives the same MMSE equalizer, where \( \tilde{d}(n) = w_{\infty, \Delta}^T \tilde{y}(n) \).

Therefore (8) can be used as an alternative cost function to (1), and the problem of equalization is converted to the problem of system modelling\(^1\) as shown in Fig. 2.

Fig. 2. The equivalent model for the MMSE equalizer.

It can be easily obtained that, when the time lag is \( \Delta \), the MMSE solution to (8) is given by:

\[
w_{\Delta} = R^{-1}(n) \cdot \mathbb{E}[y(n)\tilde{d}(n)]
= w_{\infty, \Delta N} + \delta w_{\infty, \Delta N}, \tag{9}
\]

where \( \delta w_{\infty, \Delta N} = R^{-1}(n)R_L(n)w_{\infty, \Delta L} + R^{-1}(n)R_R(n)w_{\infty, \Delta R} \), \( R(n) = \mathbb{E}[y(n)y^T(n)] \), \( R_L(n) = \mathbb{E}[y(n)y_L^T(n)] \) and \( R_R(n) = \mathbb{E}[y(n)y_R^T(n)] \). Usually the norms of \( R_L(n) \) and \( R_R(n) \) are much smaller than that of \( R(n) \), implying that \( w_{\Delta} \) is mainly determined by

\(^1\) Note that unlike other typical approaches, e.g. [12], we do not assume that channel noise is absent to obtain the *modelling* interpretation of the equalizer.
$w_{\infty,\Delta N}$ with a disturbance term of $\delta w_{\infty,\Delta N}$.

3 Optimum Time Lag

As shown in Fig. 2, the MMSE equalization is equivalent to modelling the infinitely long $w_{\infty,\Delta}(n)$ by $w(n)$ within the range of $[0, N - 1]$. Although from (5) we have $w_{\infty,\Delta} = \begin{bmatrix} w_{\infty,\Delta L}^T & w_{\infty,\Delta N}^T & w_{\infty,\Delta R}^T \end{bmatrix}^T$, only $w_{\infty,\Delta L}$ and $w_{\infty,\Delta R}$ are outside the range of $[0, N - 1]$ and contribute to the “modelling” error of $e_{\Delta}(n)$. Moreover, since $w_{\infty,\Delta}(i) = w_{\infty,0}(i - \Delta)$, the optimum time lag $\Delta_o$ shifts the most energy of $w_{\infty,\Delta}$ within the range of $[0, N - 1]$, or leaves the least energy for $w_{\infty,\Delta L}$ and $w_{\infty,\Delta R}$. To achieve a small “modelling” error, the tap-length $N$ should be long enough to cover most energy of $w_{\infty,\Delta}$. Or in other words, if $N$ is large enough, the optimum tap-length must be within $[0, N - 1]$. This can be verified following a similar procedure to that [13]. Thus usually the search for the optimum lag needs only to be constrained to within $[0, N - 1]$.

In general, the channel inverse reduces to zero for both negative and positive time; so does $w_{\infty,\Delta}(n)$. Therefore, if the tap-length $N$ is large enough, the coefficients in both $w_{\infty,\Delta L}$ and $w_{\infty,\Delta R}$ are very small when $\Delta = \Delta_o$. As $\Delta$ increases from $\Delta_o$, on the one hand, the coefficients in $w_{\infty,\Delta L}$ becomes fewer but the influence on the MMSE is trivial since they are already small; on the other hand, more coefficients are added to $w_{\infty,\Delta R}$. The overall effect is that the “modelling” error tends to increase as $\Delta$ increases from $\Delta_o$. This is also true when $\Delta$ decreases from $\Delta_o$, implying that the MMSE is generally a concave function of $\Delta$, though there may exist local minima.

With these observations, the optimum time lag $\Delta_o$ can be obtained by the gradient
descent search as:

\[ \Delta(n) = \Delta(n-1) - K \cdot \text{sign} \left( \frac{\partial \xi_\Delta(n)}{\partial \Delta(n)} \right), \]  

(10)

where \(\xi_\Delta\) is the MMSE for the time lag at \(\Delta\), \(\Delta(n)\) is the time lag at the time \(n\), \(K\) is an integer no less than 1, and “sign” is used since \(\Delta\) must be an integer. If \(K\) is large enough, (10) can escape from local minima and converge to within a range of \((\Delta_0 - K, \Delta_0 + K)\), though the algorithm can oscillate after convergence, which, like the variable step-size LMS algorithm [1,11], can be circumvented by choosing a decreasing \(K\) with respect to time. In most cases, however, we observe that the local minima are rare unless the tap-length is too short. Thus \(K\) is usually set as 1.

If \(\xi_\Delta\) for every \(\Delta\) are known, we have

\[ \frac{\partial \xi_\Delta(n)}{\partial \Delta(n)} = \frac{\xi_\Delta(n) - \xi_\Delta(n-1)}{\Delta(n) - \Delta(n-1)}. \]  

(11)

Substituting (11) into (10) gives

\[ \Delta(n) = \Delta(n-1) - K \cdot \gamma \cdot \text{sign}(\xi_\Delta(n) - \xi_\Delta(n-1)), \]  

(12)

where \(\gamma = \text{sign}(\Delta(n) - \Delta(n-1))\). The adaptation rule of (12) is the basis for the algorithm proposed later.

### 4 Time Lag Adaptation

This section proposes a novel adaptive time lag algorithm. First a new index called *pseudo fractional time lag* is introduced, on which the lag adaption is based. Then a
leaky factor is applied to prevent the time lag adaption from going to unnecessarily large values, and the tap-vector shift is presented to suppress the MSE rise at the time of lag change. Finally the over-all algorithm for the MMSE equalizer with adaptive time lag is listed, followed by a simple complexity analysis.

4.1 Pseudo Fractional Time Lag

Although theoretically the optimum lag can be searched by (12), in practice $\xi_\Delta$ is usually not available and may be estimated on-line by the exponential average as:

$$\bar{\xi}_\Delta(n) = \lambda \bar{\xi}_\Delta(n-1) + (1-\lambda)e^2(n), \quad (13)$$

where $\lambda$ is a forgetting factor close to one. Unfortunately, we can not simply substitute $\bar{\xi}_\Delta(n)$ into (12), because the time lag can only be modified by an integer value and the estimation error from (13) may cause the time lag to change dramatically, if it converges at all.

To overcome this problem, we relax the constraint that the time lag must be an integer, and introduce an index called pseudo fractional time lag which can be fractional, where the true time lag is its integer part. Then the adaptation rule of (12), with $\xi_\Delta(n)$ replaced by the estimation from (13), is applied to the fractional time lag. Only if the changes of the fractional time lag are accumulated to some extent, is the value of the true time lag changed.

\footnote{Other methods such as empirical averages, structured statistical sampling and etc may also be used for estimating the MMSE, which, however, will not affect the algorithm proposed in this section.}
To be specific, defining $\delta_f(n)$ as the pseudo fractional time lag, we have:

$$
\delta_f(n) = \delta_f(n-1) - \beta \cdot \gamma \cdot \left[\bar{\xi}_{\Delta(m)}(n) - \bar{\xi}_{\Delta(m-1)}\right],
$$

(14)

where $\gamma = \text{sign}(\Delta(m) - \Delta(m-1))$, $\bar{\xi}_{\Delta(m)}(n)$ is obtained by (13) which is the estimated MSE for the current time lag $\Delta(m)$, $\bar{\xi}_{\Delta(m-1)}$ is the stored estimated MSE for the previous lag $\Delta(m-1)$, $\beta$ is a step-size parameter, and $n$ and $m$ are the time indices for the fractional and true time lags respectively as they do not change simultaneously. Note that there is no “sign” operator in (14) since $\delta_f$ can take non-integer values, by which we have more “freedom” in adapting the time lag.

At the start (i.e $n = m = 0$), we may let $\Delta(0) = \delta_f(0) = 0$, $\Delta(-1) = -1$ and $\bar{\xi}_{\Delta(-1)} > \xi_\Delta$ for all possible $\Delta$. As the adaptation begins, both $\Delta(m)$ and $\bar{\xi}_{\Delta(m-1)}$ remain unchanged, and $\delta_f(n)$ starts to increase from 0 until at the changing time that $|\delta_f(n) - \Delta(m)| \geq K$, we increase $m$ by 1, let $\Delta(m) = \lfloor \delta_f(n) \rfloor$ which rounds $\delta_f(n)$ to the nearest integer (or equivalently $\Delta(m) = \Delta(m-1) + K$), and have $\bar{\xi}_{\Delta(m-1)} = \bar{\xi}_{\Delta(m)}(n)$.

After that, the adaptation goes on from that point. It is clear that the true time lag only varies at the changing time.

Starting at one changing time, the filter converges towards the MMSE corresponding to the new time lag. If $\beta$ is small enough, we can have $E[\bar{\xi}_{\Delta(m)}(\infty)] = \xi_{\Delta(m)}$ before the next changing time. Then taking expectations on both sides of (14) gives:

$$
E[\delta_f(n)] = E[\delta_f(n-1)] - \beta \cdot \gamma \cdot \left[\xi_{\Delta(m)} - \xi_{\Delta(m-1)}\right],
$$

(15)

\footnote{For example, we can set $\bar{\xi}_{\Delta(-1)} = 1$ if the desired signal’s power is normalized to one.}

11
where \( n \) corresponds to the converged periods of the adaptive filter. It is clear from (15) that \( E[\delta_f(n)] \) keeps increasing until:

\[
\xi_{\Delta(m)} - \xi_{\Delta(m-1)} \geq 0.
\] (16)

Similarly if we let \( \Delta(0) = \delta_f(0) > \Delta_o \), it can be verified that the time lag keeps decreasing until:

\[
\xi_{\Delta(m)} - \xi_{\Delta(m-1)} \leq 0.
\] (17)

It is clear from (16) and (17) that, if \( K \) is large enough, (14) can escape from the sub-optima and converge to a value within \((\Delta_o - K, \Delta_o + K)\) in the mean. As mentioned before, we usually set \( K = 1 \).

Rigorously speaking, (14) is a stochastic differential equation as the MMSE at time lag can only be estimated. Although the convergence behavior of a stochastic recursion should normally be analyzed in probability or in mean square, e.g. in accordance with the Kushner-Clark theorem, the detail of such analysis is beyond the scope of this paper.

4.2 Leaky Factor

If \( N \) is large, plotting MMSE versus \( \Delta \) may show a “flat line” around \( \Delta_o \). This is because, according to the analysis in Section 2.2, as the time lag moves from \( \Delta_o \) in both directions, the coefficients in both \( w_{\infty,\Delta L} \) and \( w_{\infty,\Delta R} \) may remain small until significant parts of \( w_{\infty,\Delta}(n) \) move out of the range of \([0, N - 1]\). Under such situation, rather than \( \Delta_o \), it is more desirable to set the time lag at the smallest lag among those within the “flat line”. Specifically, the desirable time lag may be defined as the smallest \( \Delta_d \) that
satisfies

\[ \xi_{\Delta} - \xi_{\Delta+K} \leq \mathcal{E} \quad \text{for all } \Delta \geq \Delta_d, \]  

(18)

where \( \mathcal{E} \) is a preset small positive value and \( K \) is a positive integer.

Due to the estimation noise, however, the lag algorithm proposed above may converge to any point within the “flat line”. Similar to the leaky LMS algorithm [1], a leaky factor \( \alpha \) is introduced in (14) to solve this problem:

\[ \delta_f(n) = (1 - \alpha) \cdot \delta_f(n-1) - \beta \gamma \left[ \bar{\xi}_{\Delta(m-1)} - \bar{\xi}_{\Delta(m)}(n) \right], \]  

(19)

where \( 0 \leq \alpha < 1 \), and to ensure stability, we should have \( 0 < \alpha \ll \gamma \). The possibility of using the leaky factor is another advantage of using the fractional time lag. Following the similar analysis in Section 4.1, we can easily verify that (19) converges in the mean to the desirable time lag defined in (18), if we let \( \alpha/\beta = \mathcal{E} \).

4.3 Tap-weight Shift

At every time that the true lag changes, there may be a sudden rise of the MSE before it eventually converges to the new MMSE, forcing us to choose a very small \( \beta \) to ensure the convergence of \( \delta_f(n) \), which, however, implies slow convergence. Thus it is necessary to estimate the optimum tap-vector for the new lag in order to suppress such a MSE rise.

It has been shown in (9) that \( \mathbf{w}_{\infty,\Delta_N} \) can be used as an estimate of the optimum tap-vector \( \mathbf{w}_\Delta \). Further noting (4) that \( w_{\infty,\Delta}(i) = w_{\infty,0}(i - \Delta) \), we have that, given the optimum tap-vector for the time lag at \( \Delta_1 \) is \( \mathbf{w}_{\Delta_1} \), the tap-vector for the time lag at
\( \Delta_2 \) can be estimated as \( \tilde{w}_{\Delta_2} = \text{SHIFT}(w_{\Delta_1}, \delta) \), where:

\[
\text{SHIFT}(w_{\Delta_1}, \delta) \triangleq \begin{cases} 
[0, \cdots, 0, w_{\Delta_1}(0), \cdots, w_{\Delta_1}(N-|\delta|-1)]^T, & \text{if } \delta > 0, \\
[w_{\Delta_1}(|\delta|), \cdots, w_{\Delta_1}(N-1), 0, \cdots, 0]^T, & \text{if } \delta < 0 
\end{cases}
\] (20)

and \( \delta_{\Delta} = \Delta_2 - \Delta_1 \). Then at each changing time that the true time lag changes, (20) is applied.

### 4.4 The Algorithm

With the above analysis, we have the algorithm for the MMSE equalizer with adaptive time lag be listed as below.

#### For every tap-input \( y(n), \ n = 1, 2, 3, \cdots \)

**Update tap-weights:**

\[
e(n) = d(n) - w^T(n)y(n)
\]

\[
w(n) = w(n-1) + (\mu/y^T(n)y(n)) \cdot e(n)y(n)
\]

**Update time lag:**

If \( n > N_k \)

\[
\xi_{\Delta(m)}(n) = \lambda \xi_{\Delta(m)}(n-1) + (1 - \lambda)e^2(n) \\
\delta_f(n) = (\delta_f(n-1) - \alpha) - \beta \gamma \left[ \xi_{\Delta(m)}(n) - \xi_{\Delta(m-1)} \right] \\
\delta_f(n) = |\delta_f(n)|_{N-1}
\]

If \( |\delta_f(n) - \Delta(n)| \geq K \)

\[
m = m + 1, \\
\Delta(m) = |\delta_f(n)|, \xi_{\Delta(m-1)} = \xi_{\Delta(m)}(n) \\
\delta_\Delta = \Delta(m) - \Delta(m-1), \gamma = \text{sign}(\delta_\Delta) \\
w(n) = \text{SHIFT}(w(n), \delta_\Delta)
\]

End

End

In the above procedure, \( \delta_f(n) = |\delta_f(n)|_{N-1} \) constrains lag adaptation within \([0, N-1]\), and to ensure stability, the lag adaptation does not begin until \( N > N_k \). Without a priori knowledge of the channel, initially we may set \( \xi_{\Delta(m-1)} = 1, \gamma = 1 \) and \( \Delta(0) = \)
\((N-1)/2\) which might be a good guess of the optimum time lag.

4.5 Complexity

The extra complexity introduced by the time lag adaption mainly comes from two parts: the MSE estimation of (13) and \(\delta_f(n)\) adaptation of (19). It can be easily verified that, at each iteration, the number of multiplications and additions are 3 and 1 for (13) respectively, and are 1 and 3 for (19) respectively. Over all, the time lag adaptation only brings about 4 extra multiplications and additions.

We note that, the normalized LMS (NLMS) algorithm requires \(2N + 1\) multiplications, \(N + 1\) additions and 1 division for one iteration. Therefore, when \(N\) is large, the extra complexity from the lag adaptation can be ignored.

5 Numerical Simulations

In this section, numerical examples are given to verify the proposed lag algorithm. We assume the transmitted signals \(x(n)\) are either +1 or −1 (BPSK) and are mutually independent, the channel SNR is 20dB, the tap-length \(N = 16\) and the step-size for the NLMS is 0.4. Based on experiments, we choose \(\beta = 0.3\), \(\alpha = 0.00025\), \(K = 1\) and \(N_k = 100\) for the time lag adaptation. Initially we let \(\Delta(0) = 7\), \(\bar{\xi}_{\Delta(m-1)} = 1\) and \(\gamma = 1\).

For clarity of exposition, all MSE learning curves below are obtained by averaging over 30 independent runs followed by a rectangular smoothing window with a size of 50, and all time lag learning curves are based on one typical simulation run. As a comparison, the MSE learning curves for the time lag fixed at \([(N-1)/2] = 7\) are also shown.
5.1 Mixed-phase Channel

In this example, the channel vector is set as $h(n) = [-0.1 \quad 0.3 \quad 0.4 \quad 1 \quad 0.4 \quad 0.3 \quad -0.1]^T$ which is a mixed-phase channel with zeros at both sides of the unit circle. The curve of the MMSE with respect to the lag is shown in Fig. 3, where we can clearly observe that the optimum lag $\Delta_o = 10$ which is within the range of $[0, N - 1]$ (note: $N = 16$).

Before simulating the proposed lag algorithm, we first examine the effect of the tap-vector estimation based on (20) in Fig. (4), where the time lag is first fixed at 6 and later increased to 7 after symbol 400. It is clear that, if the tap-vector $w(n)$ remains unchanged when the lag changes (i.e. without $w(n)$ shift), there is a sharp rise in the MSE learning curve at symbol 400, but the estimation of $w(n)$ according to (20) (i.e. with $w(n)$ shift) effectively suppresses that rise.

The MSE and time lag learning curves are shown in Fig. 5 (a) and (b) respectively. It is clearly shown in Fig. 5 (a) that the equalizer with adaptive time lag has similar convergence behavior to that with a fixed time lag at 7, but the former has lower MMSE.
Fig. 4. The effect of the tap-weight shift when the time lag changes.

because it can find the optimum time lag as shown in Fig. 5 (b).

(a) The MSE learning curve.  
(b) The time lag learning curve.

Fig. 5. Learning curves of the MSE and time lag for the mixed phase channel.
5.2 Minimum-phase and Maximum-phase Channels

This subsection shows examples for the minimum phase and maximum phase channels, where the channel vectors are set as:

\[ h(n) = [1 \ 0.8 \ 0.6 \ 0.4 \ 0.3 \ 0.2 \ 0.1]^T \]

\[ h(n) = [0.1 \ 0.2 \ 0.3 \ 0.4 \ 0.6 \ 0.8 \ 1]^T \] (21)

respectively. The curves of the MMSE versus time lag are shown in Fig. (6), where the flat line can be clearly observed in both curves. Although all time lags within the flat line correspond to similar MMSE, it is obvious from Fig. (6) that the “desirable” time lags are 0 and 15 for the minimum and maximum phase channels respectively. Clearly both 0 and 15 are in the range of \([0, N - 1]\).

Fig. 6. The MMSE vs. the time lag for the minimum phase and maximum phase channels.

The MSE and time lag learning curves for the minimum and maximum phase channels are shown in Fig. 7 and 8 respectively, where it is clearly shown that the proposed lag algorithm converges to around the ‘desirable” time lags for both channels. In Fig. (7),
though the equalizers with adaptive lag and fixed lag converge to similar MMSE level, the former has the time lag of around 0 which is obviously more appropriate than fixing the lag at 7. However, Fig. (7) (a) shows an obvious MSE rise at around symbol 500 for the proposed algorithm. This is because the first change of the time lag initially drives the search away from the desirable lag. For the maximum phase channel as shown in Fig. (8), the equalizer with adaptive time lag has significantly lower MMSE than that with fixed time lag at 7.

In general, the above results show that the equalizer with adaptive time lag can achieve better performance than that without.

![MSE learning curve](image1.png) ![Time lag learning curve](image2.png)

(a) The MSE learning curve. (b) The time lag learning curve.

Fig. 7. Learning curves of the MSE and time lag for the minimum phase channel.

6 Conclusions

This paper investigated time lag adaptation for the adaptive MMSE equalizer. First a new interpretation of the MMSE equalizer was introduced, by which the relation between the MMSE and the time lag were described. Then a novel adaptive time lag algorithm was proposed. Because the proposed algorithm is based on gradient search,
the extra complexity imposed on the NLMS algorithm is ignorable. Finally numerical simulations were given to verify the proposed algorithm.

Acknowledgements

This research was sponsored by the UK Engineering and Physical Sciences Research Council under grant number GR/S00217/01.

References


