ERROR WHITENING WIENER FILTERS: THEORY AND ALGORITHMS

Jose C. Principe, Yadunandana N. Rao, Deniz Erdogmus

Computational NeuroEngineering Laboratory
EB 451 Electrical Engineering Department
University of Florida
Gainesville, FL 32611

{principe, yadu, deniz}@cnel.ufl.edu

10.1. INTRODUCTION

The mean-squared error (MSE) criterion has been the workhorse of linear optimization theory due to the simple and analytically tractable structure of linear least squares [Farhang, 1998; Haykin, 1996]. In adaptive filter theory, the Wiener-Hopf equations are more commonly used owing to the extension of least squares to functional spaces proposed by Wiener [Farhang, 1998; Haykin, 1996]. However, for finite impulse filters (vector spaces) the two solutions coincide. There are a number of reasons behind the widespread use of the Wiener filter: firstly, the Wiener solution provides the best possible filter weights in the least squares sense; secondly, there exist simple and elegant optimization algorithms like the Least Mean Squares (LMS), Normalized Least Mean Squares (NLMS), and Recursive Least Squares (RLS) to find or closely track the Wiener solution in a sample-by-sample fashion, suitable for on-line adaptive signal processing applications [Farhang, 1998]. There are also a number of important properties that help us understand the statistical properties of the Wiener solution, namely the orthogonality of the error signal to the input vector space as well as the whiteness of the predictor error signal for stationary inputs, provided the filter is long enough [Farhang, 1998; Haykin, 1996]. However, in a number of applications of practical importance, the error sequence produced by the Wiener filter is not white. One of the most important is the case of noisy inputs. In fact, it has been long recognized that these MSE-based filter optimization approaches are unable to produce the optimal weights associated with the noise free input due to the biasing of the input covariance matrix (autocorrelation in the case of FIR filters) by the additive noise [Rao & Principe, 2002; Douglas, 1996]. Since noise is always present in real-world signals, the optimal filter weights offered by the MSE criterion and associated algorithms are inevitably inaccurate; this might hinder the performance of the designed engineering systems that require robust parameter estimations.

There are several techniques to suppress the bias in the MSE-based solutions in the presence of noisy training data [Cadzow, 1994; Lemmerling, 1999; Yeredor, 2000; So, 1999; Gao, Ahmad, & Swamy, 1994]. Total Least Squares (TLS) is one of the popular methods, due to the principled way of eliminating the effect of noise on the optimal weight vector solution [Feng, Bao, & Jiao, 1998; Golub & van Loan, 1979; Golub & van Loan, 1989]. Major drawbacks of TLS are the requirements for accurate model order estimation, an identical noise variance in the input and desired signals, and the SVD computations that severely limit its practical applicability [Golub & van Loan, 1989; Rao & Principe, 2002; de Moor, 1994; Douglas, 1996]. TLS is known to perform poorly when these assumptions are not satisfied [Yeredor, 2000; Rao & Principe, 2002]. Another important class of algorithms that can effectively eliminate noise in the input data is subspace Wiener filtering [Farhang, 1998; Haykin, 1996; Rao, 2000]. Subspace approaches try to minimize the effect of noise on the solution by projecting the input data vector onto a lower dimensional space that spans the input signal space. Traditional Wiener filtering algorithms are then applied to the projected inputs, which exhibit an improved signal-to noise ratio (SNR). Many subspace algorithms are present in the literature and it is beyond the scope of this chapter to mention all of them. The drawbacks of these methods include proper model order estimation, increased computational

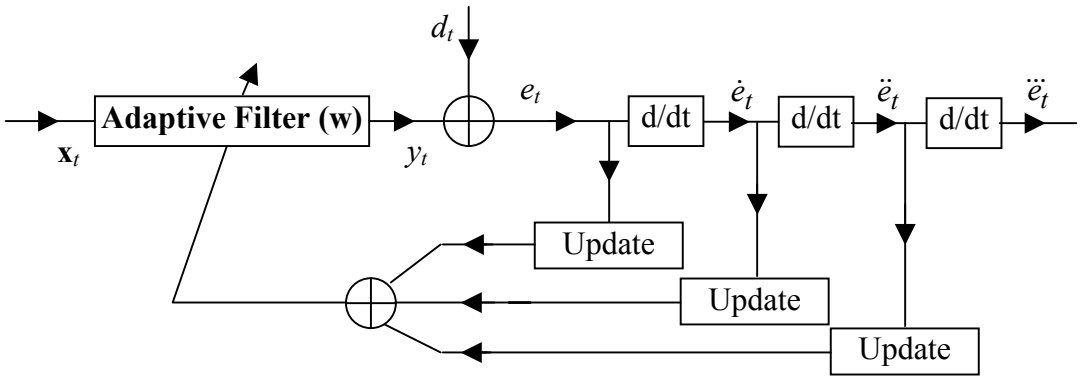


Figure 10.1. Schematic diagram of EWWF adaptation.

requirements and sufficiently small noise power that helps discriminate signal and noise during the subspace dimensionality selection [Rao, 2000].

In this chapter, we will present a completely different approach to produce a (partially) white noise sequence at the output of Wiener filters in the presence of noisy inputs. We will approach the problem by introducing a new adaptation criterion that enforces zero autocorrelation of the error signal beyond a certain lag; hence the name *error whitening Wiener filters* (EWWF). Since we want to preserve the on-line properties of the adaptation algorithms, we propose to expand the error autocorrelation around a lag larger than the filter length using Taylor series. Hence, instead of an error signal we end up with an *error vector*, with as many components as the terms kept in the Taylor series expansion. A schematic diagram of the proposed adaptation structure is depicted in Figure 10.1 The properties of this solution are very interesting, since it contains the Wiener solution as a special case, and for the case of two error terms, the same analytical tools developed for the Wiener filter can be applied with minor modifications. Moreover, when the input signal is contaminated with additive white noise, the EWWF produces the optimal solution for the noise free input signal, with the same computational complexity of the Wiener solution.

The organization of this chapter is as follows: First, we will present the motivation behind using the autocorrelation of the residual error signal in supervised training of Wiener filters. This will clearly demonstrate the reasoning behind the selected performance function, which will be called the *error whitening criterion* (EWC). Second, an analytical investigation of the mathematical properties of the EWWF and the optimal filter weight estimates will be presented. The optimal selection of parameters will be followed by demonstrations of the theoretical expectations on noise rejecting properties of the proposed solution through Monte Carlo simulations performed using analytical calculations of the necessary autocorrelation functions. Next, we will derive the Recursive Error Whitening (REW) algorithm that finds the proposed error whitening Wiener filter solution using sample-by-sample updates in a fashion similar to the well-known RLS algorithm. This type of recursive algorithms require $O(n^2)$ complexity in the number of weights. Finally, we address the issues with the development of the gradient-based algorithm for EWWF. We will derive a gradient-based LMS-type update algorithm for the weights that will converge to the vicinity of the desired solution using stochastic

updates. Theoretical bounds on step size to guarantee convergence and comparisons with MSE counter-parts will be provided.

10.2. MOTIVATION FOR ERROR WHITENING WIENER FILTERS

The classical Wiener solution yields a biased estimate of the reference filter weight vector in the presence of input noise. This problem arises due to the contamination of the input signal autocorrelation matrix with that of the additive noise. If a signal is contaminated with additive white noise, only the zero-lag autocorrelation is biased by the amount of the noise power. Autocorrelation at all other lags still remain at their original values. This observation rules out MSE as a good optimization criterion for this case. In fact, since the error power is the value of the error autocorrelation function at zero lag, the optimal weights will be biased because they depend on the input autocorrelation values at zero-lag. The fact that the autocorrelation at non-zero lags are unaffected by the presence of noise will be proved useful in determining an unbiased estimate of the filter weights.

10.2.1 Analysis of the Autocorrelation of the Error Signal

The question that arises is what lag should be used to obtain the true weight vector in the presence of white input noise. Let us consider the autocorrelation of the training error at non-zero lags. Suppose noisy training data of the form $(\mathbf{x}(t), d(t))$ is provided, where $\mathbf{x}(t) = \tilde{\mathbf{x}}(t) + \mathbf{v}(t)$ and $d(t) = \tilde{d}(t) + u(t)$ with $\tilde{\mathbf{x}}(t)$ being the sample of the noise-free input vector at time t (time is assumed to be continuous), $\mathbf{v}(t)$ being the additive white noise vector on the input vector, $\tilde{d}(t)$ being the noise-free desired output and $u(t)$ being the additive white noise on the desired output. Suppose that the true weight vector of the reference filter that generated the data is $\mathbf{w}_T$ (moving average model). Then the error at time t is $e(t) = (\tilde{d}(t) + u(t)) - (\tilde{\mathbf{x}}(t) + \mathbf{v}(t))^T \mathbf{w}$, where $\mathbf{w}$ is the estimated weight vector. Equivalently, when the desired response belongs to the subspace of the input, i.e. $\tilde{d}(t) = \tilde{\mathbf{x}}^T(t)\mathbf{w}_T$, the error can be written as

$$e(t) = (\tilde{\mathbf{x}}^T(t)\mathbf{w}_T + u(t)) - (\tilde{\mathbf{x}}(t) + \mathbf{v}(t))^T \mathbf{w} = \tilde{\mathbf{x}}^T(t)(\mathbf{w}_T - \mathbf{w}) + u(t) - \mathbf{v}^T(t)\mathbf{w} \qquad (10.1)$$

Given this noisy training data, the MSE-based Wiener solution will not yield a residual training error that has zero autocorrelation for a number of consecutive lags, even when the contaminating noise signals are white. From (10.1) it is easy to see that the error will have a zero autocorrelation function if and only if

- *the weight vector is equal to the true weights of the reference model*,
- *the lag is beyond the Wiener filter length*.

During adaptation, the issue is that the filter weights are not set at $\mathbf{w}_T$, so the error autocorrelation function will be generally nonzero. Therefore a criterion to determine the true weight vector when the data is contaminated with white noise should be *to force the*

long lags (beyond the filter length) of the error autocorrelation function to zero by using an appropriate criterion. This is exactly what the error-whitening criterion (EWC) that we propose here will do. There are two interesting situations that we should consider: What happens when the selected autocorrelation lag is smaller than the filter length? What happens when the selected autocorrelation lag is larger than the lag at which the autocorrelation function of the input signal vanishes?

The answer to the first question is simply that the solution will be still biased since it will be obtained by inverting a biased input autocorrelation matrix. If the selected lag is $L<m$ (m order of the reference filter), the bias will occur at the L^{th} sub-diagonal of the autocorrelation matrix, where the zero-lag autocorrelation of the input signal shows up. In the special case of MSE, the selected lag is zero and the zeroth sub-diagonal becomes the main diagonal, thus the solution is biased by the noise power.

The answer to the second question is practically important. The MSE solution is quite stable because it is determined by the inverse of a diagonally dominant Toeplitz matrix. The diagonal dominance is guaranteed by the fact that the autocorrelation function of a real-valued function has a peak at zero-lag. If other lags are used in the criterion, it is important that the lag is selected such that the corresponding autocorrelation matrix (which will be inverted) is not ill conditioned. If the selected lag is larger than the length of the input autocorrelation function, then the autocorrelation matrix becomes singular and a solution cannot be obtained. Therefore, lags beyond the input signal correlation time should also be avoided in practice.

10.2.2 The Structure of the Error Whitening Wiener Filters

The observation that constraining the higher lags of the error autocorrelation function to zero yields unbiased weight solutions is quite significant. Moreover, the algorithmic structure of this new solution and the lag-zero MSE solution are still very similar. The noise-free case helps us understand why this similarity occurs. Suppose the desired signal is generated by the following equation: $\widetilde{d}(t) = \widetilde{\mathbf{x}}^T(t)\mathbf{w}_T$, where $\mathbf{w}_T$ is the true weight vector. Now multiply both sides by $\widetilde{\mathbf{x}}(t-\Delta)$ from the left and then take the expected value of both sides to yield $E[\widetilde{\mathbf{x}}(t-\Delta)\widetilde{d}(t)] = E[\widetilde{\mathbf{x}}(t-\Delta)\widetilde{\mathbf{x}}^T(t)]\mathbf{w}_T$. Similarly, we can obtain $E[\widetilde{\mathbf{x}}(t)\widetilde{d}(t-\Delta)] = E[\widetilde{\mathbf{x}}(t)\widetilde{\mathbf{x}}^T(t-\Delta)]\mathbf{w}_T$. Adding the corresponding sides of these two equations yields

$$E[\widetilde{\mathbf{x}}(t)\widetilde{d}(t-\Delta) + \widetilde{\mathbf{x}}(t-\Delta)\widetilde{d}(t)] = E[\widetilde{\mathbf{x}}(t)\widetilde{\mathbf{x}}^T(t-\Delta) + \widetilde{\mathbf{x}}(t-\Delta)\widetilde{\mathbf{x}}^T(t)]\mathbf{w}_T \tag{10.2}$$

This equation is similar to the standard Wiener-Hopf equation $E[\widetilde{\mathbf{x}}(t)\widetilde{d}(t)] = E[\widetilde{\mathbf{x}}(t)\widetilde{\mathbf{x}}^T(t)]\mathbf{w}_T$. Yet, it is different due to the correlations being evaluated at a lag other than zero, which means that the weight vector can be determined by constraining higher order lags in the error autocorrelation.

10.2.3 How to Train EWWF

Now that we have described the structure of the solution, let us address the issue of training this new class of optimum filters that we called *error whitening Wiener filters* (EWWF). Adaptation exploits the sensitivity of the error autocorrelation with respect to

the weight vector of the adaptive filter. We will formulate the solution in continuous time first, for the sake of simplicity. If the support of the impulse response of the adaptive filter is of length *m*, we evaluate the derivative of the error autocorrelation function with respect to the lag Δ, where $\Delta \geq m$ are both real numbers. Assuming that the noises in the input and desired are uncorrelated to each other and the input signal, we get,

$$\begin{aligned}
\frac{\partial \rho_e(\Delta)}{\partial \mathbf{w}} &= \frac{\partial E[e(t)e(t-\Delta)]}{\partial \mathbf{w}} \\
&= \frac{\partial E\left[(\tilde{\mathbf{x}}^T(t)(\mathbf{w}_T - \mathbf{w}) + u(t) - \mathbf{v}^T(t)\mathbf{w})(\tilde{\mathbf{x}}^T(t-\Delta)(\mathbf{w}_T - \mathbf{w}) + u(t-\Delta) - \mathbf{v}^T(t-\Delta)\mathbf{w})\right]}{\partial \mathbf{w}} \\
&= \frac{\partial E\left[(\mathbf{w}_T - \mathbf{w})^T \tilde{\mathbf{x}}(t)\tilde{\mathbf{x}}^T(t-\Delta)(\mathbf{w}_T - \mathbf{w}) + (u(t) - \mathbf{v}^T(t)\mathbf{w})(u(t-\Delta) - \mathbf{v}^T(t-\Delta)\mathbf{w})\right]}{\partial \mathbf{w}} \\
&= \frac{\partial (\mathbf{w}_T - \mathbf{w})^T E\left[\tilde{\mathbf{x}}(t)\tilde{\mathbf{x}}^T(t-\Delta)\right](\mathbf{w}_T - \mathbf{w})}{\partial \mathbf{w}} \\
&= -2E\left[\tilde{\mathbf{x}}(t)\tilde{\mathbf{x}}^T(t-\Delta)\right](\mathbf{w}_T - \mathbf{w})
\end{aligned} \tag{10.3}$$

The identity in (10.3) immediately tells us that the sensitivity of the error autocorrelation with respect to the weight vector becomes zero, i.e., $\partial \rho_e(\Delta)/\partial \mathbf{w} = \mathbf{0}$, if $(\mathbf{w}_T - \mathbf{w}) = \mathbf{0}$. This observation emphasizes the following practically important conclusion: when given training data that is generated by a linear filter, but contaminated with white noise, it is possible to derive simple adaptive algorithms that could determine the underlying filter weights without bias. Furthermore, if $(\mathbf{w}_T - \mathbf{w})$ is not in the null space of $E[\tilde{\mathbf{x}}(t)\tilde{\mathbf{x}}^T(t-\Delta)]$, then only $(\mathbf{w}_T - \mathbf{w}) = \mathbf{0}$ makes $\rho_e(\Delta) = \mathbf{0}$ and $\partial \rho_e(\Delta)/\partial \mathbf{w} = \mathbf{0}$. But looking at (10.3), we conclude that a proper delay depends on the autocorrelation of the input signal that is, in general, unknown. Therefore, the selection of the delay Δ is important. One possibility is to evaluate the error autocorrelation function at different lags $\Delta \geq m$ and check for a non zero input autocorrelation function for that delay, which will be very time consuming and inappropriate for on-line algorithms.

Instead of searching for a good lag-Δ, consider the Taylor series approximation of the autocorrelation function around a *fixed lag-L*, where $L \geq m$,

$$\begin{aligned}
\rho_e(\Delta) &\approx \rho_e(L) + \dot{\rho}_e(L)(\Delta - L) + \frac{1}{2}\ddot{\rho}_e(L)(\Delta - L)^2 + \ldots \\
&= E[e(t)e(t-L)] - E[e(t)\dot{e}(t-L)](\Delta - L) + \frac{1}{2}E[e(t)\ddot{e}(t-L)](\Delta - L)^2 + \ldots
\end{aligned} \tag{10.4}$$

In (10.4), $\dot{e}(t)$ and $\ddot{e}(t)$ represent the derivatives of the error signal with respect to the time index. Notice that we do not take the Taylor series expansion around zero-lag for the reasons indicated above. Moreover, *L* should be less than the correlation time of the input, such that the Taylor expansion has a chance of being accurate. But since we bring more lags in the expansion, the choice of the lag becomes less critical than in (10.3). In principle, the more terms we keep in the Taylor expansion the more constraints we are

imposing on the autocorrelation of the error in adaptation. Therefore, instead of finding the weight vector that makes the actual gradient in (10.3) zero, we find the weight vector that makes the derivative of the approximation in (10.4) with respect to the weight vector zero.

If the adaptive filter is operating in discrete time instead of continuous time, the differentiation with respect to time can be replaced by a first-order forward difference, $\dot{e}(n) = e(n) - e(n-L)$. Higher order derivatives can also be approximated by their corresponding forward difference estimates, e.g., $\ddot{e}(n) = e(n) - 2e(n-L) + e(n-2L)$, etc. Although the forward difference normally uses two consecutive samples, for reasons that will become clear in the following sections of the chapter, we will utilize two samples separated by L samples in time. The first-order truncated Taylor series expansion for the error autocorrelation function for lag Δ evaluated at L becomes,

$$\begin{aligned} \rho_e(\Delta) &\approx E[e(n)e(n-L)] - E[e(n)(e(n) - e(n-L))](\Delta - L) \\ &= -(\Delta - L)E[e^2(n)] + (1 + \Delta - L)E[e(n)e(n-L)] \end{aligned} \tag{10.5}$$

Analyzing (10.5) we remark another advantage of the Taylor series expansion because the familiar MSE is part of the expansion. Notice also that as one forces $\Delta \to L$, the MSE term will disappear and only the lag-L error autocorrelation will remain. On the other hand, as $\Delta \to L - 1$ only the MSE term will prevail in the autocorrelation function approximation. Introducing more terms in the Taylor expansion will bring in error autocorrelation constraints from lags iL.

10.2.4 The Error Whitening Criterion

We are now in a position to formulate the error-whitening adaptation criterion. Motivated by (10.5) we designed the EWC to involve an arbitrary weighting of the two terms $e(n)$ and $\dot{e}(n)$, because yet there is no clear understanding of the trade-offs. Therefore, the EWC performance function for discrete time filtering can be written as

$$J(\mathbf{w}) = E[e^2(n)] + \beta E[\dot{e}^2(n)] \tag{10.6}$$

where β is a parameter, or equivalently

$$J(\mathbf{w}) = (1 + 2\beta)E[e^2(n)] - 2\beta E[e(n)e(n-L)] \tag{10.7}$$

which has the same form as in (10.5). The goal is to minimize $J(\mathbf{w})$ because this will enforce minimal autocorrelation at both zero and L lags. Notice that when $\beta = 0$ we recover the MSE in (10.6) and (10.7). Similarly, we would have to select $\Delta = L$ in order to make the first-order expansion identical to the exact value of the error autocorrelation function. Substituting the identity $(1 + 2\beta) = -(\Delta - L)$, and using $\Delta = L$, we observe that $\beta = -1/2$ eliminates the MSE term from the criterion. Interestingly, this value will appear in the following discussion, when we optimize β in order to reduce the bias in the solution introduced by input noise.

The same criterion can also be obtained by considering performance functions of the form

$$J(\mathbf{w}) = E\left[\left\|\left[e(n) \quad \sqrt{\beta}\dot{e}(n) \quad \sqrt{\gamma}\ddot{e}(n) \quad \ldots\right]^T\right\|_2^2\right] \\ = E[e^2(n)] + \beta E[\dot{e}^2(n)] + \gamma E[\ddot{e}^2(n)] + \ldots \tag{10.8}$$

where the coefficients β, γ, etc. are assumed to be positive. Notice that (10.8) is the L2 norm of a vector of criteria. The components of this vector consist of $e(n)$, $\dot{e}(n)$, $\ddot{e}(n)$, etc. Due to the equivalence provided by the difference approximations for derivative, these terms constrain the error autocorrelation at lags iL as well as the error power as seen in (10.8). The number of terms included in the Taylor series approximation for the error autocorrelation determines how many constraints are present in the vector of criteria. Therefore, the EWWF utilizes an *error vector* (see Figure 10.1), instead of the error signal utilized in the conventional Wiener filter. Our aim is to force the error signal as close as possible to becoming white (at lags exceeding the filter length), but these multiple lag options have not been investigated yet.

In the following sections, we will elaborate on the properties of this performance function. Specifically, we will consider the gradient (sensitivity) of (10.6) with respect to the weight vector of the adaptive filter and analyze the properties of the solution that makes this gradient equal to zero, as suggested by (10.3). It will become clear that in order to find the true weight vector of a reference filter in discrete-time operations, equating this mentioned gradient to zero will suffice. Even in the presence of noise, the true weights will be accessible by proper selection of the parameter β.

10.3. PROPERTIES OF THE ERROR WHITENING CRITERION

10.3.1 Shape of the Performance Surface

Suppose that noise-free training data of the form $(\tilde{\mathbf{x}}(n), \tilde{d}(n))$, generated by a linear system with weight vector $\mathbf{w}_T$ through $\tilde{d}(n) = \tilde{\mathbf{x}}^T(n)\mathbf{w}_T$, is provided. Assume without loss of generality that the adaptive filter and the reference filter are of the same length. This is possible since it is possible to pad $\mathbf{w}_T$ with zeros if it is shorter than the adaptive filter. Therefore, the input vector $\tilde{\mathbf{x}}(n) \in \Re^m$, the weight vector $\mathbf{w}_T \in \Re^m$ and the desired output $\tilde{d}(n) \in \Re$. The quadratic form in (10.6) defines the specific EWC we are interested in, and its unique stationary point gives the optimal solution for the EWWF. If $\beta \geq 0$, then this stationary point is a minimum. Otherwise, the Hessian of (10.6) might have mixed-sign eigenvalues or even all-negative eigenvalues. We demonstrate this fact with sample performance surfaces obtained for 2-tap FIR filters using $\beta = -1/2$. For three differently colored training data, we obtain the EWC performance surfaced shown in Figure 10.2. In each row, the MSE performance surface, the EWC cost contour plot, and the EWC performance surface are shown for the corresponding training data. The eigenvalue pairs of the Hessian matrix of (10.6) are (2.35,20.30), (-6.13,5.21), and (-4.08,-4.14), for these representative cases in Figure 10.2. Clearly, it is possible for (10.6)

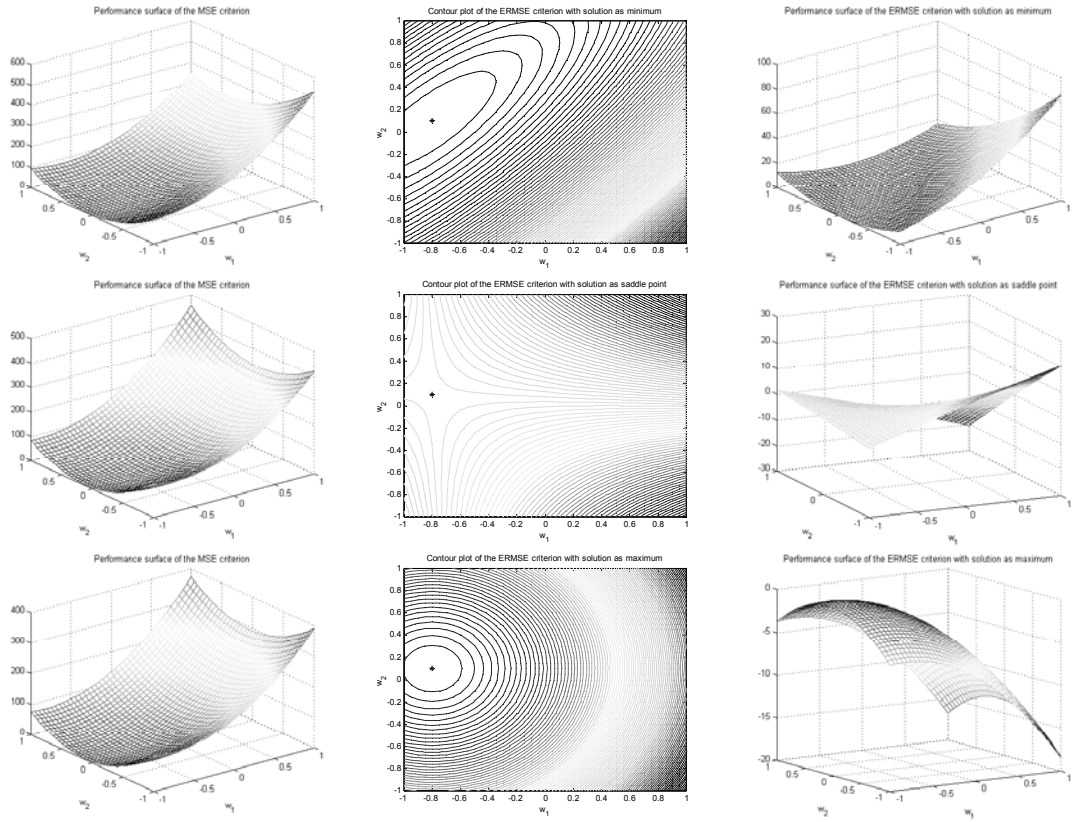


Figure 10.2. The MSE performance surfaces, the EWC contour plot, and the EWC performance surface for three different training data sets and 2-tap adaptive FIR filters.

to have a stationary point that is a minimum, a saddle point, or a maximum and we start to see the differences brought about by the EWC. The performance surface is a weighted sum of paraboloids, which will complicate gradient-based adaptation, but will not affect search algorithms utilizing curvature information.

10.3.2 Analysis of the Noise-free Input Case

Theorem 10.1. The stationary point of the quadratic form in (10.6) is given by

$$\mathbf{w}_* = (\widetilde{\mathbf{R}} + \beta\widetilde{\mathbf{S}})^{-1}(\widetilde{\mathbf{P}} + \beta\widetilde{\mathbf{Q}}) \tag{10.9}$$

where we defined $\widetilde{\mathbf{R}} = E[\widetilde{\mathbf{x}}(n)\widetilde{\mathbf{x}}^T(n)]$, $\widetilde{\mathbf{S}} = E[\dot{\widetilde{\mathbf{x}}}(n)\dot{\widetilde{\mathbf{x}}}^T(n)]$, $\widetilde{\mathbf{P}} = E[\widetilde{\mathbf{x}}(n)\widetilde{d}(n)]$, $\widetilde{\mathbf{Q}} = E[\dot{\widetilde{\mathbf{x}}}(n)\dot{\widetilde{d}}(n)]$.

Proof. Substituting the proper variables in (10.6), we obtain the following explicit expression for $J(\mathbf{w})$.

$$J(\mathbf{w}) = E[\widetilde{d}^{\,2}(n)] + \beta E[\dot{\widetilde{d}}^{\,2}(n)] + \mathbf{w}^T(\widetilde{\mathbf{R}} + \beta\widetilde{\mathbf{S}})\mathbf{w} - 2(\widetilde{\mathbf{P}} + \beta\widetilde{\mathbf{Q}})^T\mathbf{w} \tag{10.10}$$

Taking the gradient with respect to $\mathbf{w}$ and equating to zero yields

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = 2(\widetilde{\mathbf{R}} + \beta\widetilde{\mathbf{S}})\mathbf{w} - 2(\widetilde{\mathbf{P}} + \beta\widetilde{\mathbf{Q}}) = \mathbf{0}$$
$$\Rightarrow \mathbf{w}_* = (\widetilde{\mathbf{R}} + \beta\widetilde{\mathbf{S}})^{-1}(\widetilde{\mathbf{P}} + \beta\widetilde{\mathbf{Q}}) \tag{10.11}$$

Notice that selecting $\beta = 0$ in (10.6) reduces the criterion to MSE and the optimal solution, given in (10.9), reduces to the Wiener solution. Thus, the Wiener filter is a special case of the EWWF solution (though not optimal for noisy inputs, as we will show later).

Corollary 1. An equivalent expression for the stationary point of (10.6) is given by

$$\mathbf{w}_* = \left[(1+2\beta)\widetilde{\mathbf{R}} - \beta\widetilde{\mathbf{R}}_L\right]^{-1}\left[(1+2\beta)\widetilde{\mathbf{P}} - \beta\widetilde{\mathbf{P}}_L\right] \tag{10.12}$$

where we defined the matrix $\widetilde{\mathbf{R}}_L = E[\widetilde{\mathbf{x}}(n-L)\widetilde{\mathbf{x}}^T(n) + \widetilde{\mathbf{x}}(n)\widetilde{\mathbf{x}}^T(n-L)]$ and the vector $\widetilde{\mathbf{P}}_L = E[\widetilde{\mathbf{x}}(n-L)\widetilde{d}(n) + \widetilde{\mathbf{x}}(n)\widetilde{d}(n-L)]$. Notice that the interesting choice $\beta = -1/2$ yields $\mathbf{w}_* = \widetilde{\mathbf{R}}_L^{-1}\widetilde{\mathbf{P}}_L$.

Proof. Substituting the definitions of $\widetilde{\mathbf{R}}$, $\widetilde{\mathbf{S}}$, $\widetilde{\mathbf{P}}$, $\widetilde{\mathbf{Q}}$, and then recollecting terms to obtain $\widetilde{\mathbf{R}}_L$ and $\widetilde{\mathbf{P}}_L$ yields the desired result.

$$\begin{aligned}
\mathbf{w}_* &= (\widetilde{\mathbf{R}} + \beta\widetilde{\mathbf{S}})^{-1}(\widetilde{\mathbf{P}} + \beta\widetilde{\mathbf{Q}}) \\
&= \begin{Bmatrix} \left[E[\widetilde{\mathbf{x}}(n)\widetilde{\mathbf{x}}^T(n)] + \beta E[(\widetilde{\mathbf{x}}(n) - \widetilde{\mathbf{x}}(n-L))(\widetilde{\mathbf{x}}(n) - \widetilde{\mathbf{x}}(n-L))^T]\right]^{-1} \\ \left[E[\widetilde{\mathbf{x}}(n)\widetilde{d}(n)] + \beta E[(\widetilde{\mathbf{x}}(n) - \widetilde{\mathbf{x}}(n-L))(\widetilde{d}(n) - \widetilde{d}(n-L))]\right] \end{Bmatrix} \\
&= \begin{Bmatrix} \left[E[\widetilde{\mathbf{x}}(n)\widetilde{\mathbf{x}}^T(n)] + \beta(E[\widetilde{\mathbf{x}}(n)\widetilde{\mathbf{x}}^T(n)] + E[\widetilde{\mathbf{x}}(n-L)\widetilde{\mathbf{x}}^T(n-L)] - \widetilde{\mathbf{R}}_L)\right]^{-1} \\ E[\widetilde{\mathbf{x}}(n)\widetilde{d}(n)] + \beta(E[\widetilde{\mathbf{x}}(n)\widetilde{d}(n)] + E[\widetilde{\mathbf{x}}(n-L)\widetilde{d}(n-L)] - \widetilde{\mathbf{P}}_L) \end{Bmatrix} \\
&= \left[(1+2\beta)\widetilde{\mathbf{R}} - \beta\widetilde{\mathbf{R}}_L\right]^{-1}\left[(1+2\beta)\widetilde{\mathbf{P}} - \beta\widetilde{\mathbf{P}}_L\right]
\end{aligned} \tag{10.13}$$

From these results we deduct two extremely interesting conclusions:

Lemma 1. (Generalized Wiener-Hopf Equations) In the noise-free case, the true weight vector is given by $\widetilde{\mathbf{R}}_L\mathbf{w}_T = \widetilde{\mathbf{P}}_L$. (This result is also true for noisy data.)

Proof. This result follows immediately from the substitution of $\widetilde{d}(n) = \widetilde{\mathbf{x}}^T(n)\mathbf{w}_T$ and $\widetilde{d}(n-L) = \widetilde{\mathbf{x}}^T(n-L)\mathbf{w}_T$ in the definitions of $\widetilde{\mathbf{R}}_L$ and $\widetilde{\mathbf{P}}_L$.

Lemma 2. In the noise-free case, regardless of the specific value of β, the optimal solution is equal to the true weight vector, i.e., $\mathbf{w}_* = \mathbf{w}_T$.

Proof. This result follows immediately from the substitution of the result in *Lemma 1* into the optimal solution expression given in (10.9).

The result in *Lemma 1* is especially significant, since it provides a generalization of the Wiener-Hopf equations to autocorrelation and cross correlation matrices evaluated at different lags of the signals. In these equations, L represents the specific correlation lag selected, and the choice L=0 corresponds to the traditional Wiener-Hopf equations. The generalized Wiener-Hopf equations are essentially stating that, the true weight vector can be determined by exploiting correlations evaluated at different lags of the signals, and we are not restricted to the zero-lag correlations as in the Wiener solution.

10.3.3 Analysis of the Noisy Input Case

Now, suppose that we are given noisy training data $(\mathbf{x}(n), d(n))$, where $\mathbf{x}(n) = \widetilde{\mathbf{x}}(n) + \mathbf{v}(n)$ and $d(n) = \widetilde{d}(n) + u(n)$. The additive noise on both signals are zero-mean and uncorrelated with each other and with the input and desired signals. Assume that the additive noise, $u(n)$, on the desired is white (in time) and let the autocorrelation matrices of $\mathbf{v}(n)$ be $\mathbf{V} = E[\mathbf{v}(n)\mathbf{v}^T(n)]$, and $\mathbf{V}_L = E[\mathbf{v}(n-L)\mathbf{v}^T(n) + \mathbf{v}(n)\mathbf{v}^T(n-L)]$. Under these circumstances, we have to estimate the necessary matrices to evaluate (10.9) using noisy data. These matrices evaluated using noisy data, $\mathbf{R}$, $\mathbf{S}$, $\mathbf{P}$, and $\mathbf{Q}$ will become (see Appendix B for details)

$$\begin{aligned}
\mathbf{R} &= E[\mathbf{x}(n)\mathbf{x}^T(n)] = \widetilde{\mathbf{R}} + \mathbf{V} \\
\mathbf{S} &= E[(\mathbf{x}(n) - \mathbf{x}(n-L))(\mathbf{x}(n) - \mathbf{x}(n-L))^T] = 2(\widetilde{\mathbf{R}} + \mathbf{V}) - \widetilde{\mathbf{R}}_L - \mathbf{V}_L \\
\mathbf{P} &= E[\mathbf{x}(n)d(n)] = \widetilde{\mathbf{P}} \\
\mathbf{Q} &= E[(\mathbf{x}(n) - \mathbf{x}(n-L))(d(n) - d(n-L))^T] = 2\widetilde{\mathbf{P}} - \widetilde{\mathbf{P}}_L
\end{aligned} \tag{10.14}$$

Finally, the optimal solution estimate of EWC, when presented with noisy input and desired output data, will be

$$\begin{aligned}
\hat{\mathbf{w}}_* &= (\mathbf{R} + \beta\mathbf{S})^{-1}(\mathbf{P} + \beta\mathbf{Q}) \\
&= \left[(\widetilde{\mathbf{R}} + \mathbf{V}) + \beta(2(\widetilde{\mathbf{R}} + \mathbf{V}) - \widetilde{\mathbf{R}}_L - \mathbf{V}_L)\right]^{-1}\left[\widetilde{\mathbf{P}} + \beta(2\widetilde{\mathbf{P}} - \widetilde{\mathbf{P}}_L)\right] \\
&= \left[(1+2\beta)(\widetilde{\mathbf{R}} + \mathbf{V}) - \beta\widetilde{\mathbf{R}}_L - \beta\mathbf{V}_L\right]^{-1}\left[(1+2\beta)\widetilde{\mathbf{P}} - \beta\widetilde{\mathbf{P}}_L\right]
\end{aligned} \tag{10.15}$$

Theorem 10.2. (EWWF Noise-Rejection Theorem) In the noisy-input data case, the optimal solution obtained using EWC will be identically equal to the true weight vector if and only if $\beta = -1/2$, $\widetilde{\mathbf{R}}_L \neq \mathbf{0}$, and $\mathbf{V}_L = \mathbf{0}$. There are two situations to consider:

- When the adaptive linear system is an FIR filter, the input noise vector v_k consists of delayed versions of a single dimensional noise process. In that case, $\mathbf{V}_L = \mathbf{0}$ if and only if $L \geq m$, where m is the filter length and the single dimensional noise process is white.

- When the adaptive linear system is an ADALINE, the input noise is a vector process. In that case, $\mathbf{V}_L = \mathbf{0}$ if and only if the input noise vector process is white (in time) and $L \geq 1$. The input noise vector may be spatially correlated.

Proof. Sufficiency of the first statement is immediately observed by substituting the provided values of β and $\mathbf{V}_L$. Necessity is obtained by equating (10.15) to $\mathbf{w}_T$ and substituting the generalized Wiener-Hopf equations provided in *Lemma 1*. Clearly, if $\tilde{\mathbf{R}}_L = \mathbf{0}$, then there is no equation to solve, thus the weights cannot be uniquely determined using this value of *L*. The statement regarding the FIR filter case is easily proved by noticing that the temporal correlations in the noise vector diminish once the autocorrelation lag becomes greater than equal to the filter length. The statement regarding the ADALINE structure is immediately obtained from the definition of a temporally white vector process.

10.4. SOME PROPERTIES OF EWWF ADAPTATION

10.4.1 Orthogonality of Error to Input

An important question regarding the behavior of the optimal solution obtained using the EWC criterion is the relationship between the residual error signal and the input vector. In the case of MSE, we know that the Wiener solution results in the error to be orthogonal to the input signal, i.e., $E[e(n)\mathbf{x}(n)] = \mathbf{0}$ [Farhang, 1998; Haykin, 1996]. Similarly, we can determine what the EWC criterion will achieve.

Lemma 3. At the optimal solution of EWC, the error and the input random processes satisfy $\beta E[e(n)\mathbf{x}(n-L) + e(n-L)\mathbf{x}(n)] = (1+2\beta)E[e(n)\mathbf{x}(n)]$, for all $L \geq 0$.

Proof. We know that the optimal solution of EWC for any $L \geq 0$ is obtained when the gradient of the cost function with respect to the weights is zero. Therefore,

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{w}} &= 2E[e(n)\mathbf{x}(n)] + 2\beta E[(e(n) - e(n-L))(\mathbf{x}(n) - \mathbf{x}(n-L))] \\ &= (1+2\beta)E[e(n)\mathbf{x}(n)] - \beta E[e(n)\mathbf{x}(n-L) + e(n-L)\mathbf{x}(n)] = \mathbf{0} \end{aligned} \tag{10.16}$$

It is interesting to note that if $\beta = -1/2$, then we obtain $E[e(n)\mathbf{x}(n-L) + e(n-L)\mathbf{x}(n)] = \mathbf{0}$ for all *L*. On the other hand, since the criterion reduces to MSE for $\beta = 0$, then we obtain $E[e(n)\mathbf{x}(n)] = \mathbf{0}$. The result shown in (10.16), if interpreted in terms of Newtonian physics, reveals an interesting insight as to the behavior of the EWC criterion at its optimal solution (regardless of the length of the reference filter that created the desired signal). In a simplistic manner, this behavior could be summarized by the following statement: The optimal solution of EWC tries to decorrelate the residual error from the estimated future value of the input vector (see Appendix C for details).

The case where $\beta = -1/2$ is especially interesting, because it results in complete noise rejection. Notice that, in this case, since the optimal solution is equal to the true

weight vector, the residual error is given by $e(n) = u(n) - \mathbf{v}^T(n)\mathbf{w}_T$, which is composed purely of the noise in the training data. Certainly, this is the only way that the adaptive filter can achieve $E[e(n)\mathbf{x}(n-L) + e(n-L)\mathbf{x}(n)] = \mathbf{0}$ for all L values, since $E[e(n)\mathbf{x}(n-L)] = E[e(n-L)\mathbf{x}(n)] = \mathbf{0}$ for this error signal. Thus, in this special case, the EWWF not only orthogonalizes the instantaneous error and input signals, but it orthogonalizes all lags of the error from the input.

10.4.2 Relationship to Entropy Maximization

Another interesting property that the EWWF solution exhibits is its relationship with entropy. Notice that when $\beta < 0$, the optimization rule tries to minimize MSE, yet it tries to maximize the separation between samples of errors, simultaneously. We could regard the sample separation as an estimate of the error entropy. In fact, the entropy estimation literature is full of methods based on sample separations [Tarasenko, 1968; Bickel & Breiman, 1983; Hall, 1984; Beirlant & Zuijlen, 1985; Kozachenko & Leonenko, 1987; Beck & Schlogl, 1993; Tsybakov & van der Meulen, 1994]. Specifically the case $\beta = -1/2$, finds the perfect balance between entropy and MSE that allows us to eliminate the effect of noise on the solution. Recall that the Gaussian density displays maximum entropy among distributions of fixed variance. In the light of this fact, the aim of EWWF could be understood as finding the minimum error variance solution, while keeping the error close to Gaussian. Notice that, due to central limit theorem, the error signal will be closely approximated by a Gaussian density when there are a large number of taps.

10.4.3 Model Order Selection

Model order selection is another important issue in adaptive filter theory. The actual desired behavior from an adaptive filter is to find the right balance between approximating the training data as accurately as possible and generalizing to *unseen* data with precision [Bishop, 1995]. One major cause of poor generalization is known to be excessive model complexity [Bishop, 1995]. Under these circumstances, the designer's aim is to determine the least complex adaptive system (which translates to smaller number of weights in the case of linear systems) that minimizes the approximation error. Akaike's information criterion [Akaike, 1974] and Rissanen's minimum description length [Rissanen, 1989] are two important theoretical results regarding model order selection. Such methods require the designer to evaluate an objective function, which is a combination of MSE and the filter length or the filter weights, using different lengths of adaptive filters. The EWC criterion successfully determines the length of the true filter (assumed FIR), even in the presence of additive noise, provided that the trained adaptive filter is sufficiently long. In the case of an adaptive filter longer than the reference filter, the additional taps will decay to zero indicating that a smaller filter is sufficient to model the data. This is exactly what we would like an automated regularization algorithm to achieve: determining the proper length of the filter without requiring external discrete modifications on this parameter. Therefore, EWC extends the regularization capability of MSE to the case of noisy training data. Alternatively, EWC could be used as a criterion for determining the model order in a fashion similar to standard model order selection methods. Given a set of training samples, one could start solving for the optimal EWC

solution (using $\beta = -1/2$) for various lengths of the adaptive filter. As the length of the adaptive filter is increased past the length of the true filter, the error power of the EWC solution will become constant. Observing this point of transition from variable to constant error power will tell the designer exactly what the filter order should be.

10.4.4 The Effect of β on the Weight Error Vector

The effect of the cost function free parameter β on the accuracy of the solution (compared to the true weight vector that generated the training data) is another crucial issue. In fact, it is possible to determine the dynamics of the weight error as a function of β. This result is provided in the following lemma.

Lemma 4. (The Effect of β on the EWWF) In the noisy training data case, the derivative of the error vector between the optimal EWC solution and the true weight vector, i.e., $\hat{\boldsymbol{\varepsilon}}_* = \hat{\mathbf{w}}_* - \mathbf{w}_T$, with respect to β is given by

$$\frac{\partial \hat{\boldsymbol{\varepsilon}}_*}{\partial \beta} = -\left[(1+2\beta)(\mathbf{R}+\mathbf{V}) - \beta \mathbf{R}_L\right]^{-1}\left[2(\mathbf{R}-\mathbf{R}_L)\hat{\boldsymbol{\varepsilon}}_* - \mathbf{R}_L \mathbf{w}_T\right] \tag{10.17}$$

Notice that $\partial \hat{\boldsymbol{\varepsilon}}_* / \partial \beta\big|_{\beta \to -1/2} = 2\mathbf{w}_T$.

Proof. Recall from (10.15) that in the noisy data case, the optimal EWWF solution is given by $\hat{\mathbf{w}}_* = \left[(1+2\beta)(\mathbf{R}+\mathbf{V}) - \beta \mathbf{R}_L - \beta \mathbf{V}_L\right]^{-1}\left[(1+2\beta)\mathbf{P} - \beta \mathbf{P}_L\right]$. Using the chain rule for the derivative and the fact that for any nonsingular matrix $\mathbf{A}(\beta)$, $\partial \mathbf{A}^{-1} / \partial \beta = -\mathbf{A}^{-1}(\partial \mathbf{A} / \partial \beta)\mathbf{A}^{-1}$, the result in (10.17) follows from straightforward derivation. In order to get the derivative as $\beta \to -1/2$, we substitute this value and $\hat{\boldsymbol{\varepsilon}}_* = \mathbf{0}$.

The significance of *Lemma 4* is that it shows that no finite β value will make this error derivative zero. The matrix inversion, on the other hand, approaches to zero for unboundedly growing β. In addition, it could be used to determine the Euclidean error norm derivative, $\partial \left\| \hat{\boldsymbol{\varepsilon}}_* \right\|_2^2 / \partial \beta$.

10.5. NUMERICAL CASE STUDIES USING THE THEORETICAL SOLUTION

In the preceding sections, we have built the theory of the error-whitening criterion for linear adaptive filter optimization. We have investigated the behavior of the optimal solution as a function of the cost function parameters as well as determining the optimal value of this parameter in the noisy training data case. This section is designed to demonstrate these theoretical results in numerical case studies with Monte Carlo simulations. In these simulations, the following scheme will be used to generate the required autocorrelation and crosscorrelation matrices.

Given the scheme depicted in Figure 10.3, it is possible to determine the *true* analytic auto/cross-correlations of all signals of interest, in terms of the filter coefficients and the

noise powers. Suppose ξ, $\tilde{v}$, and u are zero-mean white noise signals with powers σ_x^2, σ_v^2, and σ_u^2, respectively. Suppose that the coloring filter **h** and the mapping filter **w** are unit-norm. Under these conditions, we obtain

$$E[\tilde{x}(n)\tilde{x}(n-\Delta)] = \sigma_x^2 \sum_{j=0}^{M} h_j h_{j+\Delta} \tag{10.18}$$

$$E[(\tilde{x}(n)+\tilde{v}(n))(\tilde{x}(n-\Delta)+\tilde{v}(n-\Delta))] = \begin{cases} \sigma_x^2 + \sigma_v^2, & \Delta = 0 \\ E[\tilde{x}(n)\tilde{x}(n-\Delta)], & \Delta \neq 0 \end{cases} \tag{10.19}$$

$$E[(\tilde{x}(n)+\tilde{v}(n))\hat{d}(n)] = \sigma_v^2 w_{-\Delta} + \sum_{l=0}^{N} w_l E[\tilde{x}(n)\tilde{x}(n-l-\Delta)] \tag{10.20}$$

For each combination of SNR from {-10dB,0dB,10dB}, β from {-0.5,-0.3,0,0.1}, m from {2,…,10}, and L from {m,…,20} we have performed 100 Monte Carlo simulations using randomly selected 30-tap FIR coloring and n-tap mapping filters. The length of the mapping filters and that of the adaptive filters were selected to be equal in every case. In all simulations, we used an input signal power of $\sigma_x^2 = 1$, and the noise powers $\sigma_v^2 = \sigma_u^2$ are determined from the given SNR using $SNR = 10\log_{10}(\sigma_x^2/\sigma_v^2)$. The matrices **R**, **S**, **P**, and **Q**, which are necessary to evaluate the optimal solution given by (10.15) are then evaluated using (10.18), (10.19), and (10.20), analytically. The results obtained are summarized in Figure 10.4 and Figure 10.5, where for the three SNR levels selected, the average squared error norm for the optimal solutions (in reference to the true weights) are given as a function of L and n for different β values. In Figure 10.4, we present the average normalized weight vector error norm obtained using EWC at different SNR levels and using different β values as a function of the correlation lag L that is used in the criterion. The filter length is 10 in these results. From the theoretical analysis, we know that if the input autocorrelation matrix is invertible, then the solution accuracy should be independent of the autocorrelation lag L. The results of the Monte Carlo simulations presented in Figure 10.4 conform to this fact. As expected, the optimal choice of $\beta = -1/2$ determined the correct filter weights exactly.

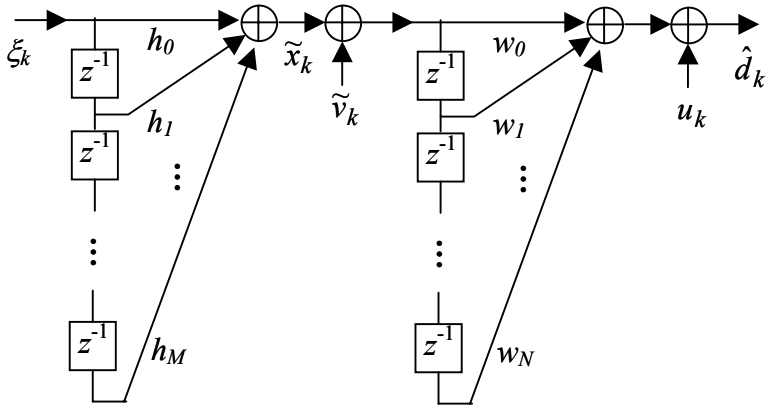


Figure 10.3. Demonstration scheme with coloring filter **h**, true mapping filter **w**, and the uncorrelated white signals ξ, $\tilde{v}$, and $\hat{u}$.

Another set of results, presented in Figure 10.5, shows the effect of filter length on the accuracy of the solutions provided by the EWC criterion. The optimal value of $\beta = -1/2$ always yields the perfect solution, whereas the accuracy of the optimal weights degrades as this parameter is increased towards zero (i.e. as the weights approaches the Wiener solution). An interesting observation from Figure 10.5 is that for SNR levels below zero, the accuracy of the solutions using sub-optimal β values increases, whereas for SNR levels above zero, the accuracy decreases when the filter length is increased. For zero SNR, on the other hand, the accuracy seems to be roughly unaffected by the filter length.

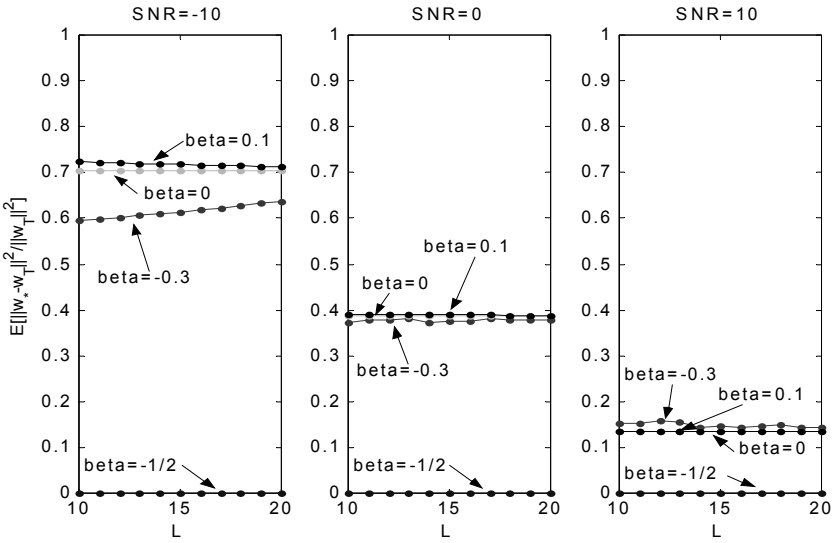


Figure 10.4. The average squared error-norm of the optimal weight vector as a function of autocorrelation lag L for various β values and SNR levels.

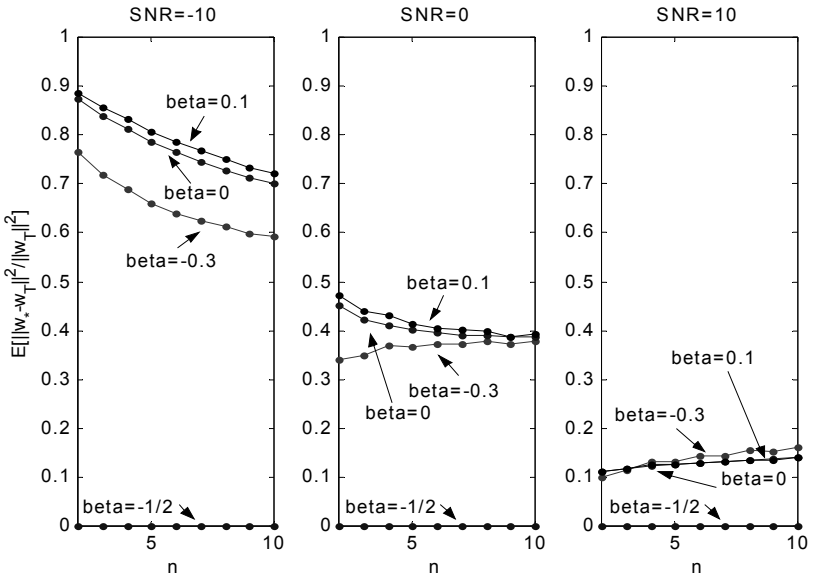


Figure 10.5. The average squared error-norm of the optimal weight vector as a function of filter length m for various β values and SNR levels.

The Monte Carlo simulations performed in the preceding examples utilized the exact coloring filter and the true filter coefficients to obtain the analytical solutions. In our final case study, we demonstrate the performance of the batch solution of the EWC criterion obtained from sample estimates of all the relevant auto- and cross-correlation matrices. In these Monte Carlo simulations, we utilize 10,000 samples corrupted with white noise at various SNR levels. The results of these Monte Carlo simulations are summarized in the histograms shown in Figure 10.6. Each subplot of Figure 10.6 corresponds to experiments performed using SNR levels of –10 dB, 0dB, and 10 dB for each column and adaptive filter lengths of 4-taps, 8-taps, and 12-taps for each row, respectively. For each combination of SNR and filter length, we have performed 50 Monte Carlo simulations using MSE ($\beta = 0$) and EWC ($\beta = -1/2$) criteria. The correlation lag is selected to be equal to the filter length in all simulations, due to *Theorem 10.2*. Clearly, Figure 10.6 demonstrates the superiority of the EWWF in rejecting noise that is present in the training data. Notice that in all subplots (i.e., for all combinations of filter length and SNR), EWWF achieves a smaller average error norm than MSE. The discrepancy between the performances of the two solutions intensifies with increasing filter length. Next, we demonstrate the error-whitening property of the proposed EWC solutions.

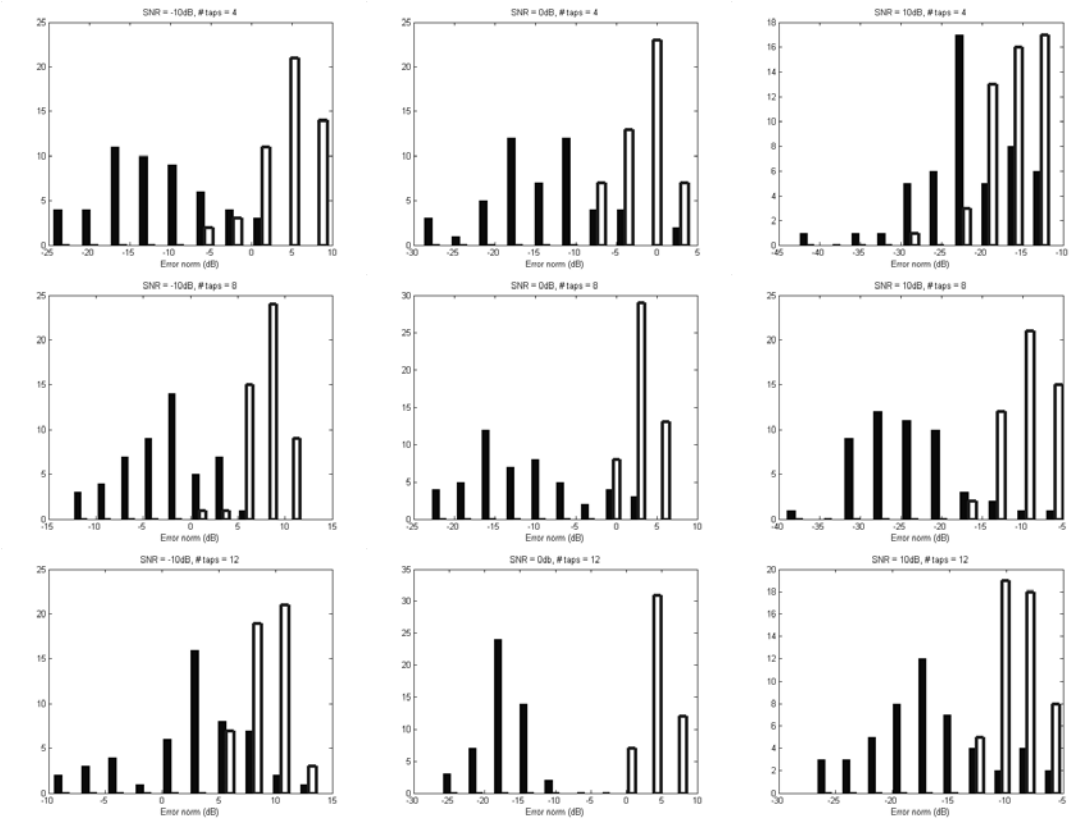


Figure 10.6. Histograms of the weight error norms (dB) obtained in 50 Monte Carlo simulations using 10000 samples of noisy data using MSE (empty bars) and EWC with $\beta = -1/2$ (full bars). The subfigures in each row use filters with 4, 8, and 12 taps respectively. The subfigures in each column use noisy samples at –10, 0, and 10 dB SNR, respectively.

From (10.1) we can expect that the error autocorrelation function will vanish at lags greater than or equal to the length of the reference filter, if the weight vector is identical to the true weight vector. For any other value of the weight vector, the error autocorrelation fluctuates at non-zero values. A 4-tap reference filter is identified with a 4-tap adaptive filter using noisy training data (hypothetical) at an SNR level of 0dB. The autocorrelation functions of the error signals corresponding to the MSE solution and the EWC solution are shown in Figure 10.7. Clearly, the EWC criterion determines a solution that forces the error autocorrelation function to zero at lags greater than or equal the filter length (partial whitening of the error).

Finally, we address the order selection capability and demonstrate how the EWC criterion can be used as a tool for determining the correct filter order, even with noisy data, provided that the given input-desired output pair is a moving average process. For this purpose, we determine the theoretical Wiener and EWWF (with $\beta = -1/2$ and $L = m$, where m is the length of the adaptive filter) solutions for a randomly selected pair of coloring filter, **h**, and mapping filter **w**, at different adaptive filter lengths. The noise level is selected to be 20 dB, and the length of the true mapping filter is 5. We know from our theoretical analysis that if the adaptive filter is longer than the reference filter, the EWWF will yield the true weight vector padded with zeros. This will not change the MSE of the solution. Thus, if we plot the MSE of the EWWF versus the length of the adaptive filter, starting from the length of the actual filter, the MSE curve will remain flat, whereas the Wiener solution will keep decreasing the MSE, contaminating the solution by learning the noise in the data. Figure 10.8a shows the MSE of the Wiener solution as well as the EWWF obtained for different lengths of the adaptive filter using the same training data described above. Notice (in the zoomed-in portion) that the MSE of the EWWF remains constant starting from 5, which is the filter order that generated the data. On the other hand, if we were to decide on the filter order looking at the MSE of the Wiener solution, we would select a model order of 4, since the gain in MSE is insignificantly small compared to the previous steps from this point on.

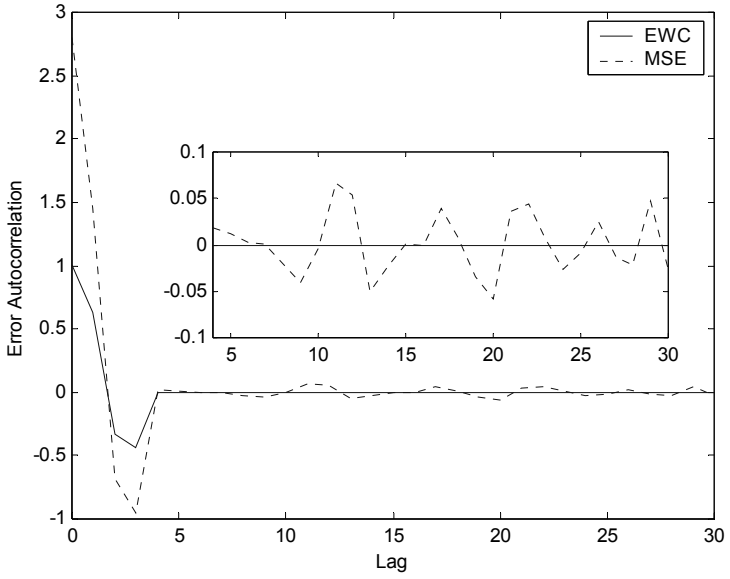


Figure 10.7. Error autocorrelation function for MSE (dotted) and EWC (solid) solutions.

Figure 10.8b shows the norm of the weight vector error for the solutions obtained using the EWC and MSE criteria, which confirms that the true weight vector is indeed attained with the EWC criterion once the proper model order is reached.

This section aimed at experimentally demonstrating the theoretical concepts set forth in the preceding sections of the paper. We have demonstrated with numerous Monte Carlo simulations that the analytical solution of the EWC criterion eliminates the effect of noise completely if the proper value is used for β. We have also demonstrated that the batch solution of EWC (estimated from a finite number of samples) outperforms MSE in the presence of noise, provided that a sufficient number of samples are given so that the noise autocorrelation matrices diminish as required by the theory.

Although we have presented a complete theoretical investigation of the proposed criterion and its analytical solution, in practice, on-line algorithms that operate on a sample-by-sample basis to determine the desired solution are equally valuable. Therefore, in the sequel, we will focus on designing computationally efficient on-line algorithms to solve for the EWWF in a fashion similar to the well-known LMS and RLS algorithms. In fact, we aim to come up with algorithms that have the same computational complexity with these two widely used algorithms. The advantage of these new algorithms will be their ability to provide better estimates of the model weights when the training data is contaminated with white noise.

10.6. THE RECURSIVE ERROR WHITENING (REW) ALGORITHM

In this section, we will present an on-line recursive algorithm to estimate the optimal solution for the error-whitening criterion. Given the estimate of the filter tap weights at time instant $(n-1)$ the goal is to determine the best set of tap weights at the next iteration n that would track the optimal solution. This algorithm, which we call Recursive Error Whitening (REW), is similar to the Recursive Least-Squares (RLS). The strongest motivation behind proposing the REW algorithm is that it is truly a fixed-point type algorithm that tracks, at each iteration, the optimal solution.

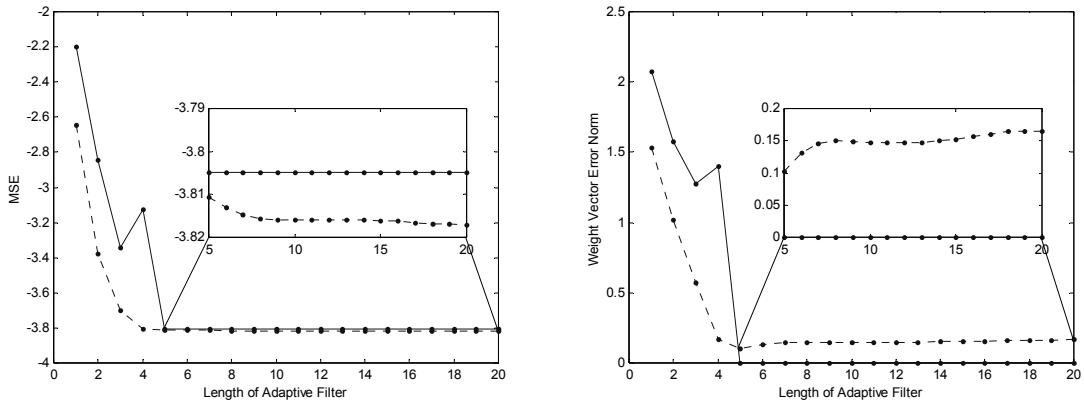


Figure 10.8. Model order selection using the EWC criterion; a) MSE - $E[d^2(n)]$ of the EWWF (solid) and the Wiener solutions (dotted) versus filter length, b) Norm of the weight vector error as a function of filter length for EWWF (solid) and Wiener solutions (dotted).

This tracking nature results in the faster convergence of the REW algorithm [Regalia, 1995]. This, however, comes at an increase in the computational cost. The REW algorithm is $O(m^2)$ in complexity (same as the RLS algorithm) and this is a substantial increase in the complexity when compared with simple gradient methods that will be discussed in a later section. We know that the optimal solution for the error-whitening criterion is given by,

$$\mathbf{w}_* = \left(\mathbf{R} + \beta\mathbf{S}\right)^{-1}\left(\mathbf{P} + \beta\mathbf{Q}\right) \tag{10.21}$$

Letting $\mathbf{T}(n) = \mathbf{R}(n) + \beta\mathbf{S}(n)$ and $\mathbf{V}(n) = \mathbf{P}(n) + \beta\mathbf{Q}(n)$, we obtain the following recursions

$$\begin{aligned}\mathbf{T}(n) &= \mathbf{T}(n-1) + (1+2\beta)\mathbf{x}(n)\mathbf{x}^T(n) - \beta\mathbf{x}(n-L)\mathbf{x}^T(n) - \beta\mathbf{x}(n)\mathbf{x}^T(n-L) \\ &= \mathbf{T}(n-1) + 2\beta\mathbf{x}(n)\mathbf{x}^T(n) - \beta\mathbf{x}(n-L)\mathbf{x}^T(n) + \mathbf{x}(n)\mathbf{x}^T(n) - \beta\mathbf{x}(n)\mathbf{x}^T(n-L) \\ &= \mathbf{T}(n-1) + (2\beta\mathbf{x}(n) - \beta\mathbf{x}(n-L))\mathbf{x}^T(n) + \mathbf{x}(n)(\mathbf{x}(n) - \beta\mathbf{x}(n-L))^T\end{aligned} \tag{10.22}$$

The well known Sherman-Morrison-Woodbury identity or the matrix inversion lemma [Golub & van Loan, 1979] states,

$$(\mathbf{A} + \mathbf{BCD}^T)^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1} + \mathbf{D}^{\mathbf{T}}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{D}^{\mathbf{T}}\mathbf{A}^{-1} \tag{10.23}$$

Substituting $\mathbf{A} = \mathbf{T}(n-1)$, $\mathbf{B} = [(2\beta\mathbf{x}(n) - \beta\mathbf{x}(n-L)) \quad \mathbf{x}(n)]$, $\mathbf{C} = \mathbf{I}_{2x2}$, a 2x2 identity matrix and $\mathbf{D} = [\mathbf{x}(n) \quad (\mathbf{x}(n) - \beta\mathbf{x}(n-L))]$, we see that (10.22) is obtained. Therefore, the recursion for the inverse of $\mathbf{T}(n)$ becomes

$$\mathbf{T}^{-1}(n) = \mathbf{T}^{-1}(n-1) - \mathbf{T}^{-1}(n-1)\mathbf{B}(\mathbf{I}_{2x2} + \mathbf{D}^T\mathbf{T}^{-1}(n-1)\mathbf{B})^{-1}\mathbf{D}^T\mathbf{T}^{-1}(n-1) \tag{10.24}$$

Note that the computation of the above inverse is different (and more involved) than the conventional RLS algorithm. It requires the inversion of an extra 2x2 matrix $(\mathbf{I}_{2x2} + \mathbf{D}^T\mathbf{T}^{-1}(n-1)\mathbf{B})$. The recursive estimator for $\mathbf{V}(n)$ is a simple correlation estimator given by,

$$\mathbf{V}(n) = \mathbf{V}(n-1) + [(1+2\beta)d(n)\mathbf{x}(n) - \beta d(n)\mathbf{x}(n-L) - \beta d(n-L)\mathbf{x}(n)] \tag{10.25}$$

Using $\mathbf{T}^{-1}(n)$ and $\mathbf{V}(n)$, an estimate of the filter weight vector at iteration index *n* is

$$\mathbf{w}(n) = \mathbf{T}^{-1}(n)\mathbf{V}(n) \tag{10.26}$$

We will define a gain matrix analogous to the gain vector in the RLS case [Haykin, 1996] as,

$$\boldsymbol{\kappa}(n) = \mathbf{T}^{-1}(n-1)\mathbf{B}\left(\mathbf{I}_{2x2} + \mathbf{D}^T\mathbf{T}^{-1}(n-1)\mathbf{B}\right)^{-1} \tag{10.27}$$

Using the above definition, the recursive estimate for the inverse of $\mathbf{T}(n)$ becomes,

$$\mathbf{T}^{-1}(n) = \mathbf{T}^{-1}(n-1) - \boldsymbol{\kappa}(n)\mathbf{D}^T\mathbf{T}^{-1}(n-1) \tag{10.28}$$

Once again, the above equation is analogous to the Ricatti equation for the RLS algorithm. Multiplying (10.27) from the right by $(\mathbf{I}_{2x2} + \mathbf{D}^T\mathbf{T}^{-1}(n-1)\mathbf{B})$, we obtain,

$$\begin{aligned}
&\boldsymbol{\kappa}(n)\left(\mathbf{I}_{2x2} + \mathbf{D}^T\mathbf{T}^{-1}(n-1)\mathbf{B}\right) = \mathbf{T}^{-1}(n-1)\mathbf{B} \\
&\boldsymbol{\kappa}(n) = \mathbf{T}^{-1}(n-1)\mathbf{B} - \boldsymbol{\kappa}(n)\mathbf{D}^T\mathbf{T}^{-1}(n-1)\mathbf{B} \\
&\boldsymbol{\kappa}(n) = \mathbf{T}^{-1}(n)\mathbf{B}
\end{aligned} \tag{10.29}$$

In order to derive an update equation for the filter weights, we substitute the recursive estimate for $\mathbf{V}(n)$ in (10.26).

$$\mathbf{w}(n) = \mathbf{T}^{-1}(n)\mathbf{V}(n-1) + \mathbf{T}^{-1}(n)[(1+2\beta)d(n)\mathbf{x}(n) - \beta d(n)\mathbf{x}(n-L) - \beta d(n-L)\mathbf{x}(n)] \tag{10.30}$$

Using (10.28) and recognizing the fact that $\mathbf{w}(n-1) = \mathbf{T}^{-1}(n-1)\mathbf{V}(n-1)$ the above equation can be reduced to,

$$\begin{aligned}
\mathbf{w}(n) &= \mathbf{w}(n-1) - \boldsymbol{\kappa}(n)\mathbf{D}^T\mathbf{w}(n-1) \\
&\quad + \mathbf{T}^{-1}(n)[(1+2\beta)d(n)\mathbf{x}(n) - \beta d(n)\mathbf{x}(n-L) - \beta d(n-L)\mathbf{x}(n)]
\end{aligned} \tag{10.31}$$

Using the definition for $\mathbf{B} = [(2\beta\mathbf{x}(n) - \beta\mathbf{x}(n-L)) \quad \mathbf{x}(n)]$, we can easily see that,

$$(1+2\beta)d(n)\mathbf{x}(n) - \beta d(n)\mathbf{x}(n-L) - \beta d(n-L)\mathbf{x}(n) = \mathbf{B}\begin{bmatrix} d(n) \\ d(n) - \beta d(n-L) \end{bmatrix} \tag{10.32}$$

From (10.29) and (10.32), the weight update equation simplifies to,

$$\mathbf{w}(n) = \mathbf{w}(n-1) - \boldsymbol{\kappa}(n)\mathbf{D}^T\mathbf{w}(n-1) + \boldsymbol{\kappa}(n)\begin{bmatrix} d(n) \\ d(n) - \beta d(n-L) \end{bmatrix} \tag{10.33}$$

Note that the product $\mathbf{D}^T\mathbf{w}(n-1)$ is nothing but the matrix of the outputs $\begin{bmatrix} y(n) & y(n) - \beta y(n-L) \end{bmatrix}^T$, where $y(n) = \mathbf{x}^T(n)\mathbf{w}(n-1)$, $y(n-L) = \mathbf{x}^T(n-L)\mathbf{w}(n-1)$. The apriori error matrix is defined as,

$$\mathbf{e}(n) = \begin{bmatrix} d(n) - y(n) \\ d(n) - y(n) - \beta(d(n-L) - y(n-L)) \end{bmatrix} = \begin{bmatrix} e(n) \\ e(n) - \beta e(n-L) \end{bmatrix} \tag{10.34}$$

Using all the above definitions, we will formally state the weight update equation for the REW algorithm as,

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \boldsymbol{\kappa}(n)\mathbf{e}(n) \tag{10.35}$$

The overall complexity of (10.35) is $O(m^2)$ which is comparable to the complexity of the RLS algorithm. Unlike the stochastic gradient algorithms that are easily affected by the eigenspread of the input data and the type of the stationary point solution (minimum, maximum or saddle), the REW algorithm is immune to these problems. This is because it inherently makes use of more information about the performance surface by computing the inverse of the Hessian matrix $\mathbf{R} + \beta\mathbf{S}$. A summary of the REW algorithm is given below in Table 10.I.

Table 10.I: Summary of the REW algorithm

Initialize $\mathbf{T}^{-1}(0) = c\mathbf{I}$, c is a large positive constant

$\mathbf{w}(0) = \mathbf{0}$

At every iteration, compute

$\mathbf{B} = [(2\beta\mathbf{x}(n) - \beta\mathbf{x}(n-L)) \quad \mathbf{x}(n)]$ and $\mathbf{D} = [\mathbf{x}(n) \quad (\mathbf{x}(n) - \beta\mathbf{x}(n-L))]$

$\boldsymbol{\kappa}(n) = \mathbf{T}^{-1}(n-1)\mathbf{B}\left(\mathbf{I}_{2x2} + \mathbf{D}^T\mathbf{T}^{-1}(n-1)\mathbf{B}\right)^{-1}$

$y(n) = \mathbf{x}^T(n)\mathbf{w}(n-1)$ and $y(n-L) = \mathbf{x}^T(n-L)\mathbf{w}(n-1)$

$$\mathbf{e}(n) = \begin{bmatrix} d(n) - y(n) \\ d(n) - y(n) - \beta(d(n-L) - y(n-L)) \end{bmatrix} = \begin{bmatrix} e(n) \\ e(n) - \beta e(n-L) \end{bmatrix}$$

$\mathbf{w}(n) = \mathbf{w}(n-1) + \boldsymbol{\kappa}(n)\mathbf{e}(n)$

$\mathbf{T}^{-1}(n) = \mathbf{T}^{-1}(n-1) - \boldsymbol{\kappa}(n)\mathbf{D}^T\mathbf{T}^{-1}(n-1)$

The convergence analysis of the REW algorithm is similar to the analysis of the RLS algorithm, which is dealt in detail in Haykin [Haykin, 1996]. In this article, we will not dwell further on the convergence issues of REW algorithm. The REW algorithm as given by (10.35) works for the stationary data only. For non-stationary data, tracking becomes an important issue and this can be handled by including a forgetting factor in the estimation of $\mathbf{T}^{-1}(n)$ and $\mathbf{V}(n)$. This generalization of the REW algorithm with forgetting factor is trivial and very similar to the exponentially weighted RLS (EWRLS) algorithm [Haykin, 1996].

10.6.1 Estimation of System Parameters in White Noise Using REW

The REW algorithm can be used effectively to solve the system identification problem in noisy environments. As we have seen before, setting the value of $\beta = -0.5$, noise immunity can be gained for parameter estimation. We generated a purely white

Gaussian random noise of length 50,000 samples and added this to a colored input signal. The white noise signal is uncorrelated with the input signal. The noise free, colored, input signal was filtered by the unknown reference filter, and this formed the desired signal for the adaptive filter. Since, the noise in the desired signal would be averaged out for both RLS and REW algorithms, we decided to use the clean desired signal itself. This will bring out only the effects of input noise on the filter estimates. Also, the noise added to the clean input is uncorrelated with the desired signal. In the experiment, we varied the Signal-to-Noise-Ratio (SNR) in the range –10dB to +10dB. The number of desired filter coefficients was also varied from 4 to 12. We then performed 100 Monte Carlo runs and computed the normalized error vector norm given by,

$$error = 20\log 10\left[\frac{\left\|\mathbf{w}_T - \mathbf{w}_*\right\|}{\left\|\mathbf{w}_T\right\|}\right] \tag{10.36}$$

where, $\mathbf{w}_*$ is the weight vector estimated by the REW algorithm with $\beta = -0.5$ after 50,000 iterations or one complete presentation of the input data and $\mathbf{w}_T$ is the true weight vector. In order to show the effectiveness of the REW algorithm, we performed Monte Carlo runs using the RLS algorithm on the same data to estimate the filter coefficients. Figure 10.9 shows a histogram plot of the normalized error vector norm given in (10.36). The solid bars show the REW results and the empty bars denote the results of RLS. It is clear that the REW algorithm is able to perform better than the RLS at various SNR and tap length settings. In the high SNR cases, there is not much of a difference between RLS and REW results. However, under noisy circumstances, the reduction in the parameter estimation error with REW is orders of magnitude more when compared with RLS. Also, the RLS algorithm results in a rather useless zero weight vector, i.e., $\mathbf{w} = \mathbf{0}$ when the SNR is lower than –10dB. It is rather well known that the RLS algorithm results in a biased estimate of the filter parameters in the presence of noisy input signals [Douglas, 1995].

10.6.2 Effect of β and Weight Tracks of REW Algorithm

Since we have a free parameter β to choose, it would be worthwhile to explore the effect of β on the parameter estimates. The SNR of the input signal is fixed at 0dB and –10dB, the number of filter taps is set to 4 and the desired signal is noise free as before. We performed 100 Monte Carlo experiments and analyzed the average error vector norm values for $-1 \le \beta \le 1$. The results of the experiment are shown in Figure 10.10. Notice that there is a dip at $\beta = -0.5$ (indicated by a "*" in the figure) and this clearly gives us the minimum estimation error. For $\beta = 0$, (indicated by a "o" in the figure) the REW algorithm reduces to the regular RLS giving a fairly significant estimation error. Next the parameter β is set to –0.5 and SNR to 0dB, and the weight tracks are estimated for the two algorithms. Figure 10.11 shows the averaged weight tracks for both REW and RLS algorithms over 50 Monte Carlo trials. Asterisks on the plots indicate the true parameters. The tracks for the RLS algorithm are smoother, but they converge to wrong values, which we have observed quite consistently. The weight tracks for the REW algorithm are

noisier compared to those of the RLS, but they eventually converge to values very close to the true weights.

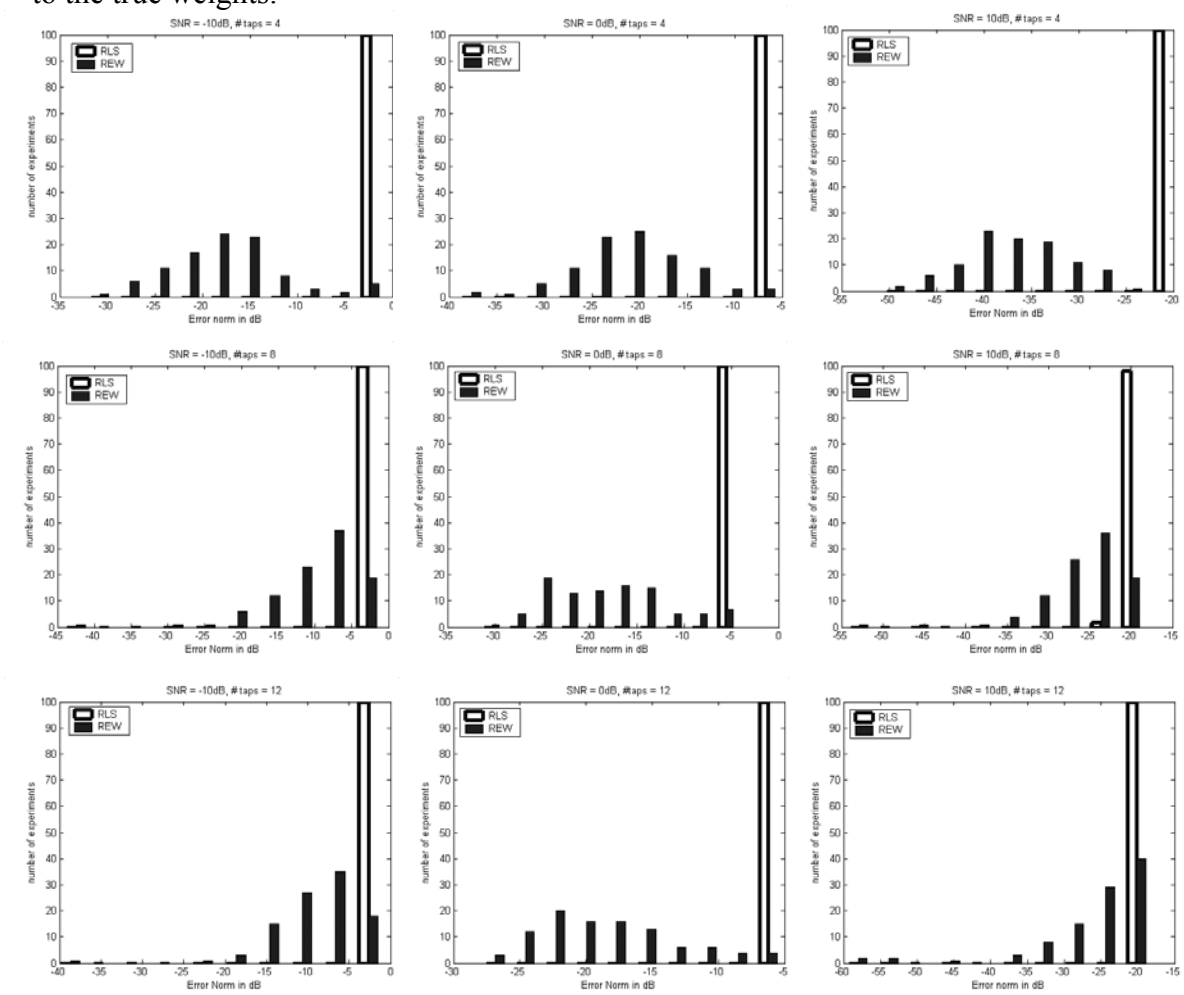
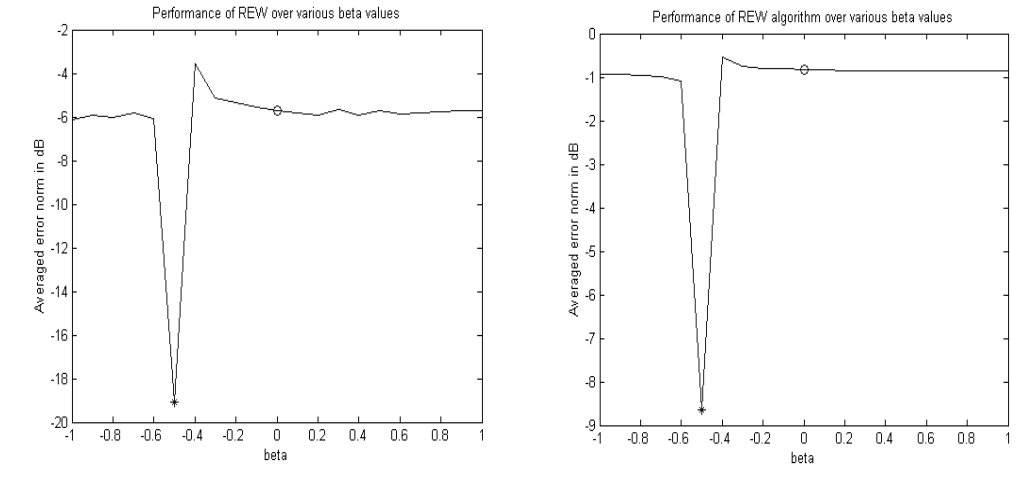
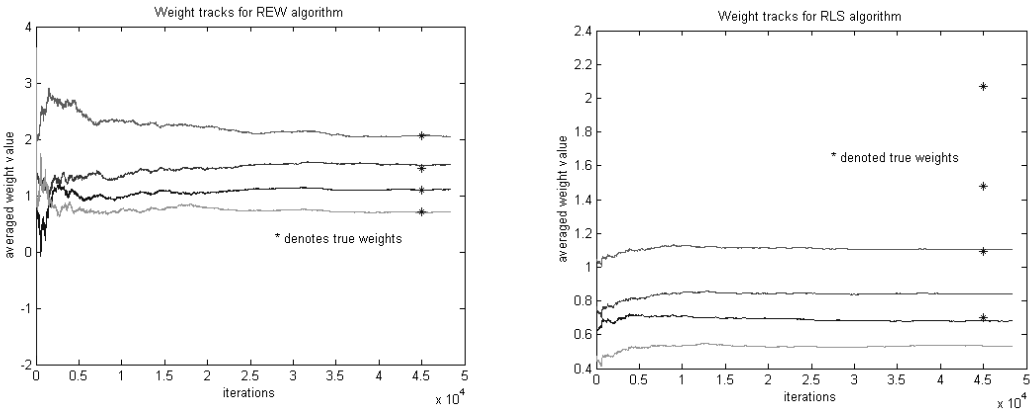


Figure 10.9. Histogram plots showing the normalized error vector norm for REW and RLS algorithms.



(a) (b)

Figure 10.10. Performance of REW algorithm (a) SNR = 0dB and (b) SNR = -10 over



various beta values.

Figure 10.11. Weight tracks for REW and RLS algorithms.

We have observed that the weight tracks for the REW algorithm can be quite noisy in the initial stages of adaptation. This may be attributed to the ill conditioning that is mainly caused by the smallest eigenvalue of the estimated Hessian matrix, which is $\mathbf{R}(n) + \beta\mathbf{S}(n)$ for the REW algorithm. The same holds true for the RLS algorithm, where the minimum eigenvalue of $\mathbf{R}(n)$ affects the sensitivity [Haykin, 1996]. The instability issues of the RLS algorithm during the initial stages of adaptation have been well studied in literature and effects of round off error have been analyzed and many solutions have been proposed to make the RLS algorithm robust to such effects [Haykin, 1996; Mueller, 1981; Chansarkar & Desai, 1997]. Similar analysis on the REW algorithm is yet to be done and this would be addressed in future work on the topic.

10.7. STOCHASTIC GRADIENT ALGORITHMS

Stochastic gradient algorithms have been at the forefront in optimizing quadratic cost functions like the MSE. Owing to the presence of a global minimum in quadratic performance surfaces, gradient algorithms can elegantly accomplish the task of reaching the optimal solution at minimal computational cost. In this section, we will derive the stochastic gradient algorithms for the EWC. Since the EWC performance surface is a weighted sum of quadratics, we can expect that difficulties will arise. Assume that we have a noisy training data set of the form $(\mathbf{x}(n), d(n))$, where $\mathbf{x}(n) \in \Re^m$ is the input and $d(n) \in \Re$ is the output of a linear system with coefficient vector $\mathbf{w}_T$. The goal is to estimate the parameter vector $\mathbf{w}_T$ using the EWC. We know that the EWC cost function is given by,

$$J(\mathbf{w}) = E[e^2(n)] + \beta E[\dot{e}^2(n)] \tag{10.37}$$

where, $\dot{e}(n) = e(n) - e(n-L)$, $\mathbf{w}$ is the estimate of the parameter vector and $L \geq m$, the size of the input vector. For convenience, we will restate the following definitions, $\dot{\mathbf{x}}(n) = \mathbf{x}(n) - \mathbf{x}(n-L)$, $\dot{d}(n) = d(n) - d(n-L)$, $\mathbf{R} = E[\mathbf{x}(n)\mathbf{x}^T(n)]$, $\mathbf{S} = E[\dot{\mathbf{x}}(n)\dot{\mathbf{x}}^T(n)]$, $\mathbf{P} = E[\mathbf{x}(n)d(n)]$ and $\mathbf{Q} = E[\dot{\mathbf{x}}(n)\dot{d}(n)]$. Using these definitions, we can rewrite the cost function in (10.37) as,

$$J(\mathbf{w}) = E[d^2(n)] + \beta E[\dot{d}^2(n)] + \mathbf{w}^T(\mathbf{R} + \beta\mathbf{S})\mathbf{w} - 2(\mathbf{P} + \beta\mathbf{Q})^T\mathbf{w} \tag{10.38}$$

It is easy to see that both $E[e^2(n)]$ and $E[\dot{e}^2(n)]$ have parabolic performance surfaces as their Hessians have positive eigenvalues. However, the value of β can invert the performance surface of $E[\dot{e}^2(n)]$. For $\beta > 0$, the stationary point is always a global minimum and the gradient of (10.38) can be written as the sum of the individual gradients as shown below.

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = 2(\mathbf{R} + \beta\mathbf{S})\mathbf{w} - 2(\mathbf{P} + \beta\mathbf{Q}) = 2(\mathbf{R}\mathbf{w} - \mathbf{P}) + 2\beta(\mathbf{S}\mathbf{w} - \mathbf{Q}) \tag{10.39}$$

The above gradient can be approximated by the stochastic instantaneous gradient by removing the expectation operators,

$$\frac{\partial J(\mathbf{w}(n))}{\partial \mathbf{w}(n)} \approx e(n)\mathbf{x}(n) + \beta\dot{e}(n)\dot{\mathbf{x}}(n) \tag{10.40}$$

Thus we can write the weight update for the stochastic EWC-LMS algorithm for $\beta > 0$ as,

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta(n)\left(e(n)\mathbf{x}(n) + \beta\dot{e}(n)\dot{\mathbf{x}}(n)\right) \tag{10.41}$$

where $\eta(n) > 0$ is a finite step-size parameter that controls convergence. For $\beta < 0$, the stationary point is still unique, but it can be a saddle point, global maximum or a global minimum. Evaluating the gradient as before and using the instantaneous gradient, we get the EWC-LMS algorithm for $\beta < 0$,

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta(n)\left(e(n)\mathbf{x}(n) - |\beta|\dot{e}(n)\dot{\mathbf{x}}(n)\right) \tag{10.42}$$

where, $\eta(n)$ is again a small step-size. However, there is no guarantee that the above update rules will be stable for all choices of step-sizes. Although, equations (10.41) and (10.42) are identical, we will use $-|\beta|$ in the update equation (10.42) to analyze the convergence of the algorithm specifically for $\beta < 0$. The reason for the separate analysis is that the convergence characteristics of (10.41) and (10.42) are very different.

Theorem 10.3. The stochastic EWC algorithms asymptotically converge in the mean to the optimal solution given by

$$\begin{aligned} \mathbf{w}_* &= \left(\mathbf{R} + \beta\mathbf{S}\right)^{-1}\left(\mathbf{P} + \beta\mathbf{Q}\right), \quad \beta > 0 \\ \mathbf{w}_* &= \left(\mathbf{R} - |\beta|\mathbf{S}\right)^{-1}\left(\mathbf{P} - |\beta|\mathbf{Q}\right), \quad \beta < 0 \end{aligned} \tag{10.43}$$

We will make the following mild assumptions typically applied to stochastic gradient algorithms [Haykin, 1996; Ljung, 1977; Kushner & Clark, 1978; Benveniste, Metivier, & Priouret, 1990] that can be easily satisfied.

A.1 The input vectors $\mathbf{x}(n)$ are derived from at least a wide sense stationary (WSS) colored random signal with positive definite autocorrelation matrix $\mathbf{R} = E[\mathbf{x}(n)\mathbf{x}^T(n)]$

A.2 The matrix $\mathbf{R}_L = E[\mathbf{x}(n)\mathbf{x}^T(n-L) + \mathbf{x}(n-L)\mathbf{x}^T(n)]$ exists and has full rank

A.3 The sequence of weight vectors $\mathbf{w}(n)$ is bounded with probability 1

A.4 The update functions $h(\mathbf{w}(n)) = e(n)\mathbf{x}(n) + \beta\dot{e}(n)\dot{\mathbf{x}}(n)$ for $\beta > 0$ and $h(\mathbf{w}(n)) = e(n)\mathbf{x}(n) - |\beta|\dot{e}(n)\dot{\mathbf{x}}(n)$ for $\beta < 0$ exist and are continuously differentiable with respect to $\mathbf{w}(n)$, and their derivatives are bounded in time.

A.5 Even if $h(\mathbf{w}(n))$ has some discontinuities a mean update vector $\bar{h}(\mathbf{w}(n)) = \lim_{n\to\infty} E[h(\mathbf{w}(n))]$ exists.

Assumption *A.1* is easily satisfied. *A.2* requires that the input signal have sufficient correlation with itself for at least L lags.

10.7.1 Proof of EWC-LMS Convergence for $\beta > 0$

We will first consider the update equation in (10.41) which is the stochastic EWC-LMS algorithm for $\beta > 0$. Without loss of generality, we will assume that the input vectors $\mathbf{x}(n)$ and their corresponding desired responses $d(n)$ are noise-free. The mean update vector $\bar{h}(\mathbf{w}(n))$ is given by,

$$\bar{h}(\mathbf{w}(n)) = \frac{d\mathbf{w}(t)}{dt} = E[e(n)\mathbf{x}(n) + \beta\dot{e}(n)\dot{\mathbf{x}}(n)] = \mathbf{R}\mathbf{w}(n) - \mathbf{P}(n) + \beta\left(\mathbf{S}\mathbf{w}(n) - \mathbf{Q}(n)\right) \tag{10.44}$$

The stationary point of the ordinary differential equation (ODE) in (10.44) is given by,

$$\mathbf{w}_* = \left(\mathbf{R} + \beta\mathbf{S}\right)^{-1}\left(\mathbf{P} + \beta\mathbf{Q}\right) \tag{10.45}$$

We will define the error vector at time instant n as $\boldsymbol{\xi}(n) = \mathbf{w}_* - \mathbf{w}(n)$. Therefore,

$$\boldsymbol{\xi}(n+1) = \boldsymbol{\xi}(n) - \eta(n)[e(n)\mathbf{x}(n) + \beta\dot{e}(n)\dot{\mathbf{x}}(n)] \tag{10.46}$$

and the norm of the error vector at time n+1 is simply,

$$\begin{aligned}\|\xi(n+1)\|^2 &= \|\xi(n)\|^2 - 2\eta(n)[\xi^T(n)e(n)\mathbf{x}(n) + \beta\xi^T(n)\dot{e}(n)\dot{\mathbf{x}}(n)] \\ &\quad + \eta^2(n)\|e(n)\mathbf{x}(n) + \beta\dot{e}(n)\dot{\mathbf{x}}(n)\|^2\end{aligned} \tag{10.47}$$

Imposing the condition that $\|\xi(n+1)\|^2 < \|\xi(n)\|^2$ for all n, we get an upper bound on the time varying step-size parameter $\eta(n)$ which is given by,

$$\eta(n) < \frac{2[\xi^T(n)e(n)\mathbf{x}(n) + \beta\xi^T(n)\dot{e}(n)\dot{\mathbf{x}}(n)]}{\|e(n)\mathbf{x}(n) + \beta\dot{e}(n)\dot{\mathbf{x}}(n)\|^2} \tag{10.48}$$

Simplifying the above equation using the fact that $\xi^T(n)\mathbf{x}(n) = e(n)$ and $\xi^T(n)\dot{\mathbf{x}}(n) = \dot{e}(n)$, we get

$$\eta(n) < \frac{2[e^2(n) + \beta\dot{e}^2(n)]}{\|e(n)\mathbf{x}(n) + \beta\dot{e}(n)\dot{\mathbf{x}}(n)\|^2} \tag{10.49}$$

which is a more practical upper bound on the step-size as it can be directly estimated from the input and outputs. As an observation, we would like to say that if $\beta = 0$, then, the bound in (10.49) reduces to,

$$\eta(n) < \frac{2}{\|\mathbf{x}(n)\|^2} \tag{10.50}$$

which, when included in the update equation, reduces to a variant of the Normalized LMS (NLMS) algorithm. In general, if the step-size parameter is chosen according to the bound given by (10.49), then the norm of the error vector $\xi(n)$ is a monotonically decreasing sequence converging asymptotically to zero, i.e., $\lim_{n\to\infty}\|\xi(n)\|^2 \to 0$ which implies that $\lim_{n\to\infty}\mathbf{w}(n) \to \mathbf{w}_*$. In addition, the upper bound on the step-size ensures that the weights are always bound with probability one satisfying the assumption *A.3* we made before. Thus the weight vector $\mathbf{w}(n)$ converges asymptotically to $\mathbf{w}_*$, which is the only stable stationary point of the ODE in (10.44). Note that (10.41) is an $O(m)$ algorithm.

10.7.2 Proof of EWC-LMS Convergence for $\beta < 0$

We analyze the convergence of the stochastic gradient algorithm for $\beta < 0$ in the presence of white noise because this is the relevant case ($\beta = -0.5$ eliminates the bias due to noise added to the input). From (10.42), the mean update vector $\bar{h}(\mathbf{w}(n))$ is given by,

$$\bar{h}(\mathbf{w}(n)) = \frac{d\mathbf{w}(t)}{dt} = E[e(n)\mathbf{x}(n) - |\beta|\dot{e}(n)\dot{\mathbf{x}}(n)] = \mathbf{R}\mathbf{w}(n) - \mathbf{P}(n) - |\beta|\left(\mathbf{S}\mathbf{w}(n) - \mathbf{Q}(n)\right) \quad (10.51)$$

As before, the stationary point of this ODE is,

$$\mathbf{w}_* = \left(\mathbf{R} - |\beta|\mathbf{S}\right)^{-1}\left(\mathbf{P} - |\beta|\mathbf{Q}\right) \quad (10.52)$$

The eigenvalues of $\mathbf{R} - |\beta|\mathbf{S}$ decide the nature of the stationary point. If they are all positive, then we have a global minimum and if they are all negative, we have a global maximum. In these two cases, the stochastic gradient algorithm in (10.42) with proper fixed sign step-size would converge to the stationary point, which would be stable. However, we know that the eigenvalues of $\mathbf{R} - |\beta|\mathbf{S}$ can also take both positive and negative values resulting in a saddle stationary point. Thus, the underlying dynamical system would have both stable and unstable modes making it impossible for the algorithm in (10.42) with fixed sign step-size to converge. This is well known in the literature [Haykin, 1994]. However, as will be shown next, this difficulty can be removed for our case by appropriately utilizing the sign of the update equation (remember that this saddle point is the only stationary point of the quadratic performance surface). The general idea is to use a vector step-size (one stepsize per weight) having both *positive and negative values*. One unrealistic way (for an on-line algorithm) to achieve this goal is to estimate the eigenvalues of $\mathbf{R} - |\beta|\mathbf{S}$. Alternatively, we can derive the conditions on the step-size for guaranteed convergence. As before, we will define the error vector at time instant n as $\boldsymbol{\xi}(n) = \mathbf{w}_* - \mathbf{w}(n)$. The norm of the error vector at time instant n+1 is given by,

$$\begin{aligned}\|\boldsymbol{\xi}(n+1)\|^2 &= \|\boldsymbol{\xi}(n)\|^2 - 2\eta(n)[\boldsymbol{\xi}^T(n)e(n)\mathbf{x}(n) - |\beta|\boldsymbol{\xi}^T(n)\dot{e}(n)\dot{\mathbf{x}}(n)] \\ &\quad + \eta^2(n)\|e(n)\mathbf{x}(n) - |\beta|\dot{e}(n)\dot{\mathbf{x}}(n)\|^2\end{aligned} \quad (10.53)$$

Taking the expectations on both sides, we get,

$$\begin{aligned}E\|\boldsymbol{\xi}(n+1)\|^2 &= E\|\boldsymbol{\xi}(n)\|^2 - 2\eta(n)E[\boldsymbol{\xi}^T(n)e(n)\mathbf{x}(n) - |\beta|\boldsymbol{\xi}^T(n)\dot{e}(n)\dot{\mathbf{x}}(n)] \\ &\quad + \eta^2(n)E\|e(n)\mathbf{x}(n) - |\beta|\dot{e}(n)\dot{\mathbf{x}}(n)\|^2\end{aligned} \quad (10.54)$$

The mean of the error vector norm will monotonically decay to zero over time i.e., $E\|\boldsymbol{\xi}(n+1)\|^2 < E\|\boldsymbol{\xi}(n)\|^2$ if and only if the step-size satisfies the following inequality.

$$|\eta(n)| < \frac{2\left|E[\boldsymbol{\xi}^T(n)e(n)\mathbf{x}(n) - |\beta|\boldsymbol{\xi}^T(n)\dot{e}(n)\dot{\mathbf{x}}(n)]\right|}{E\|e(n)\mathbf{x}(n) - |\beta|\dot{e}(n)\dot{\mathbf{x}}(n)\|^2} \quad (10.55)$$

Let $\mathbf{x}(n) = \tilde{\mathbf{x}}(n) + \mathbf{v}(n)$ and $d(n) = \tilde{d}(n) + u(n)$, where $\tilde{\mathbf{x}}(n)$ and $\tilde{d}(n)$ be the clean input and desired data respectively. We will further assume that the input noise vector $\mathbf{v}(n)$ and the noise component in the desired signal $u(n)$ to be uncorrelated. Also the noise signals are assumed to be independent of the clean input and desired signals. Furthermore, the lag L is chosen to be more than *m*, the length of the filter under consideration. Since the noise is assumed to be purely white, $E[\mathbf{v}(n)\mathbf{v}^T(n-L)] = E[\mathbf{v}(n-L)\mathbf{v}^T(n)] = 0$ and $E[\mathbf{v}(n)\mathbf{v}^T(n)] = \mathbf{V}$. We have,

$$\boldsymbol{\xi}^T(n)e(n)\mathbf{x}(n) = \left(\mathbf{w}_* - \mathbf{w}(n)\right)^T \left(\tilde{d}(n) + u(n) - \mathbf{w}^T(n)\tilde{\mathbf{x}}(n) - \mathbf{w}^T(n)\mathbf{v}(n)\right)\left(\tilde{\mathbf{x}}(n) + \mathbf{v}(n)\right) \tag{10.56}$$

Simplifying this further and taking the expectations, we get,

$$\begin{aligned} E[\boldsymbol{\xi}^T(n)e(n)\mathbf{x}(n)] &= \text{var}(\tilde{d}(n)) - 2\tilde{\mathbf{P}}^T\mathbf{w}(n) + \mathbf{w}^T(n)\tilde{\mathbf{R}}\mathbf{w}(n) \\ &\quad + \mathbf{w}^T(n)\mathbf{V}\mathbf{w}(n) - \mathbf{w}_*^T\mathbf{V}\mathbf{w}(n) \\ &= J_{MSE} - \mathbf{w}_*^T\mathbf{V}\mathbf{w}(n) \end{aligned} \tag{10.57}$$

where, $\tilde{\mathbf{R}} = E[\tilde{\mathbf{x}}(n)\tilde{\mathbf{x}}^T(n)]$, $\tilde{\mathbf{P}} = E[\tilde{\mathbf{x}}(n)\tilde{d}(n)]$ and

$$J_{MSE} = \mathbf{w}^T(n)\left(\tilde{\mathbf{R}} + \mathbf{V}\right)\mathbf{w}(n) + \text{var}(\tilde{d}(n)) - 2\tilde{\mathbf{P}}^T\mathbf{w}(n) \tag{10.58}$$

Similarly, we have,

$$\begin{aligned} \boldsymbol{\xi}^T(n)\dot{e}(n)\dot{\mathbf{x}}(n) &= \left(\mathbf{w}_* - \mathbf{w}(n)\right)^T[\tilde{d}(n) + u(n) - \mathbf{w}^T(n)(\tilde{\mathbf{x}}(n) + \mathbf{v}(n)) \\ &\quad - \tilde{d}(n-L) - u(n-L) + \mathbf{w}^T(n)(\tilde{\mathbf{x}}(n-L) + \mathbf{v}(n-L))]\cdot \\ &\quad \left(\tilde{\mathbf{x}}_k + \mathbf{v}_k - \tilde{\mathbf{x}}_{k-L} - \mathbf{v}_{k-L}\right) \end{aligned} \tag{10.59}$$

Evaluating the expectations on both sides of (10.59) and simplifying, we obtain,

$$\begin{aligned} E[\boldsymbol{\xi}^T(n)\dot{e}(n)\dot{\mathbf{x}}(n)] &= \text{var}(\tilde{d}(n) - \tilde{d}(n-L)) - 2\tilde{\mathbf{Q}}^T\mathbf{w}(n) \\ &\quad + \mathbf{w}^T(n)\tilde{\mathbf{S}}\mathbf{w}(n) + 2\mathbf{w}^T(n)\mathbf{V}\mathbf{w}(n) - 2\mathbf{w}_*^T\mathbf{V}\mathbf{w}(n) \\ &= J_{ENT} - 2\mathbf{w}_*^T\mathbf{V}\mathbf{w}(n) \end{aligned} \tag{10.60}$$

where, we have used the definitions $\tilde{\mathbf{S}} = E[(\tilde{\mathbf{x}}(n) - \tilde{\mathbf{x}}(n-L))(\tilde{\mathbf{x}}(n) - \tilde{\mathbf{x}}(n-L))^T]$, $\tilde{\mathbf{Q}} = E[(\tilde{\mathbf{x}}(n) - \tilde{\mathbf{x}}(n-L))(\tilde{d}(n) - \tilde{d}(n-L))]$ and

$$J_{ENT} = \mathbf{w}^T(n)\left(\tilde{\mathbf{S}} + 2\mathbf{V}\right)\mathbf{w}(n) + \text{var}(\tilde{d}(n) - \tilde{d}(n-L)) - 2\tilde{\mathbf{Q}}^T\mathbf{w}(n) \tag{10.61}$$

Using (10.57) and (10.60) in equation (10.55), we get an expression for the upper bound on the step-size as,

$$|\eta(n)| < \frac{2\left|J_{MSE} - |\beta| J_{ENT} - (1-2|\beta|)\mathbf{w}_*^T \mathbf{V}\mathbf{w}(n)\right|}{E\left\|e(n)\mathbf{x}(n) - |\beta|\dot{e}(n)\dot{\mathbf{x}}(n)\right\|^2} \tag{10.62}$$

This expression is not usable in practice as an upper bound because it depends on the optimal weight vector. However, for $\beta = -0.5$, the upper bound on the step-size reduces to,

$$|\eta(n)| < \frac{2\left|J_{MSE} - 0.5 J_{ENT}\right|}{E\left\|e(n)\mathbf{x}(n) - 0.5\dot{e}(n)\dot{\mathbf{x}}(n)\right\|^2} \tag{10.63}$$

From (10.58) and (10.61), we know that J_{MSE} and J_{ENT} are positive quantities. However, $J_{MSE} - 0.5 J_{ENT}$ can be negative. Also note that this upper bound is computed by evaluating the right hand side of (10.63) with the current weight vector $\mathbf{w}(n)$. Thus as expected, it is very clear that the step-size at the n^{th} iteration can take either positive or negative values based on $J_{MSE} - 0.5 J_{ENT}$; therefore, $\text{sgn}(\eta(n))$ must be the same as $\text{sgn}(J_{MSE} - 0.5 J_{ENT})$ evaluated at $\mathbf{w}(n)$. Intuitively speaking, the term $J_{MSE} - 0.5 J_{ENT}$ is the EWC cost computed with the current weights $\mathbf{w}(n)$ and $\beta = -0.5$, which tells us where we are on the performance surface and the sign tells which way to go to reach the stationary point. It also means that the lower bound on the step-size is not positive as in traditional gradient algorithms. In general, if the step-size we choose satisfies (10.62), then, the mean error vector norm decreases asymptotically i.e., $E\left\|\boldsymbol{\xi}(n+1)\right\|^2 < E\left\|\boldsymbol{\xi}(n)\right\|^2$ and eventually becomes zero, which implies that $\lim_{n\to\infty} E[\mathbf{w}(n)] \to \mathbf{w}_*$. Thus the weight vector $E[\mathbf{w}(n)]$ converges asymptotically to $\mathbf{w}_*$, which is the only stationary point of the ODE in (10.51). We conclude that the knowledge of the eigenvalues is not needed to implement gradient descent in the EWC performance surface, but (10.63) is still not appropriate for a simple LMS type algorithm.

10.7.3 On-line Implementations of EWC-LMS for $\beta < 0$

As mentioned before, computing $J_{MSE} - 0.5 J_{ENT}$ at the current weight vector would require reusing the entire past data at every iteration. As an alternative, we can extract the curvature at the operating point and include that information in the gradient algorithm. By doing so, we obtain the following stochastic algorithm

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta\,\text{sgn}\left(\mathbf{w}^T(n)\left[\mathbf{R}(n) - |\beta|\mathbf{S}(n)\right]\mathbf{w}(n)\right)\left(e(n)\mathbf{x}(n) - |\beta|\dot{e}(n)\dot{\mathbf{x}}(n)\right) \tag{10.64}$$

where, $\mathbf{R}(n)$ and $\mathbf{S}(n)$ are the estimates of $\mathbf{R}$ and $\mathbf{S}$ respectively at the n^{th} time instant.

Corollary: Given any quadratic surface $J(\mathbf{w})$, the following gradient algorithm converges to its stationary point.

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \eta \operatorname{sgn}(\mathbf{w}^T(n)\mathbf{H}\mathbf{w}(n)) \frac{\partial J}{\partial \mathbf{w}(n)} \tag{10.65}$$

Proof: Without loss of generality, suppose that we are given a quadratic surface of the form $J(\mathbf{w}) = \mathbf{w}^T\mathbf{H}\mathbf{w}$, where $\mathbf{H} \in \Re^{m\times m}$, and $\mathbf{w} \in \Re^{mx1}$. $\mathbf{H}$ is restricted to be symmetric; therefore, it is the Hessian matrix of this quadratic surface. The gradient of the performance surface with respect to the weights, evaluated at point $\mathbf{w}_0$ is $\frac{\partial J}{\partial \mathbf{w}_0} = 2\mathbf{H}\mathbf{w}_0$, and the stationary point of $J(\mathbf{w})$ is the origin. Since the performance surface is quadratic, any cross-section passing through the stationary point is a parabola. Consider the cross-section of $J(\mathbf{w})$ along the line defined by the local gradient that passes through the point $\mathbf{w}_0$. In general, the Hessian matrix of this surface can be positive or negative definite; it might as well have mixed eigenvalues. The unique stationary point of $J(\mathbf{w})$, which makes its gradient zero, can be reached by moving along the direction of the local gradient. The important issue is the selection of the sign, i.e., whether to move along or against the gradient direction to reach the stationary point. The decision can be made by observing the local curvature of the cross-section of $J(\mathbf{w})$ along the gradient direction. The performance surface cross-section along the gradient direction at $\mathbf{w}_0$ is,

$$J(\mathbf{w}_0 + 2\eta\mathbf{H}\mathbf{w}_0) = \mathbf{w}_0^T(I + 2\eta\mathbf{H})^T\mathbf{H}(I + 2\eta\mathbf{H})\mathbf{w}_0 = \mathbf{w}_0^T(\mathbf{H} + 4\eta\mathbf{H}^2 + 4\eta^2\mathbf{H}^3)\mathbf{w}_0 \tag{10.66}$$

From this, we deduce that the local curvature of the parabolic cross-section at $\mathbf{w}_0$ is $4\mathbf{w}_0^T\mathbf{H}^3\mathbf{w}_0$. If the performance surface is locally convex, then this curvature is positive. If the performance surface is locally concave, this curvature is negative. Also, note that $\operatorname{sgn}(4\mathbf{w}_0^T\mathbf{H}^3\mathbf{w}_0) = \operatorname{sgn}(\mathbf{w}_0^T\mathbf{H}\mathbf{w}_0)$. Thus, the update equation with the curvature information in (10.65) converges to the stationary point of the quadratic cost function $J(\mathbf{w})$ irrespective of the nature of the stationary point.

From the above corollary and utilizing the fact that the matrix $\mathbf{R} - |\beta|\mathbf{S}$ is symmetric, we can conclude that the update equation in (10.64) asymptotically converges to the stationary point $\mathbf{w}_* = (\mathbf{R} - |\beta|\mathbf{S})^{-1}(\mathbf{P} - |\beta|\mathbf{Q})$. On the down side however, the update equation in (10.64) requires $O(m^2)$ computations, which makes the algorithm unwieldy for real-world applications. Also, we can use the REW algorithm instead, which has a similar complexity.

For an $O(m)$ algorithm, we have to go back to the update rule in (10.42). We will discuss only the simple case of $\beta = -0.5$, which turns out to be also the more useful. We propose to use an instantaneous estimate of the sign with the current weights given by

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta(n)\,\mathrm{sgn}(e^2(n) - 0.5\dot{e}^2(n))[e(n)\mathbf{x}(n) - 0.5\dot{e}(n)\dot{\mathbf{x}}(n)] \qquad (10.67)$$

where, where $\eta(n) > 0$ and is bound by (10.63). It is possible to make mistakes in the sign estimation when (10.67) is utilized, which will not affect the convergence in the mean, but will penalize the misadjustment. The argument that misadjustment will be more for the EWC algorithm in (10.67) than the standard LMS algorithm is currently under investigation.

10.7.4 Estimation of System Parameters in White Noise

The experimental setup is the same as the one we used to test the REW algorithm. We varied the Signal-to-Noise-Ratio (SNR) between –10dB to +10dB and changed the number of filter parameters from 4 to 12. We set $\beta = -0.5$ and used the update equation in (10.67) for the EWC-LMS algorithm. A time varying step-size magnitude was chosen in accordance with the upper bound given by (10.63) without the expectation operators. This greatly reduces the computational burden but makes the algorithm noisier. However, since we are using 50,000 samples for estimating the parameters, we can expect the errors to average out over iterations. For the LMS algorithm, we chose the step-size that gave the least error in each trial. Totally 100 Monte Carlo trials were performed and histograms of normalized error vector norms were plotted. Figure 10.12 shows the error histograms for both LMS and EWC-LMS algorithms. EWC-LMS algorithm performs significantly better than the LMS algorithm at low SNR values. Their performances are on par for SNRs greater than 20dB. Figure 10.13 shows a sample comparison between the stochastic and the recursive algorithms for 0dB SNR and 4 filter taps. Interestingly, the performance of the EWC-LMS algorithm is better than the REW algorithm in the presence of noise. Similarly, the LMS algorithm is much better than the RLS algorithm. This tells us that the stochastic algorithms reject more noise than the fixed-point algorithms. Researchers have made this observation before, although no concrete arguments exist to account for the smartness of the adaptive algorithms [Reuter, Quirk, Zeidler, & Milstein, 2000]. Similar conclusions can be drawn in our case for EWC-LMS and REW.

10.7.5 Weight Tracks and Convergence

The steady state performance of a stochastic gradient algorithm is a matter of great importance. We will now experimentally verify the steady state behavior of the EWC-LMS algorithm. The SNR of the input signal is set to 10dB and the number of filter taps is fixed to two for display convenience. Figure 10.14 shows the contour plot of the EWC cost function with noisy input data. Clearly, the Hessian of this performance surface has both positive and negative eigenvalues thus making the stationary point an undesirable saddle point. On the same plot, we have shown the weight tracks of the EWC-LMS algorithm in (10.67) with $\beta = -0.5$. Also, we used a fixed value of 0.001 for the step-size. From the figure, it is clear that the EWC-LMS algorithm converges stably to the saddle point solution, which is theoretically unstable when a single sign step-size is used. Notice that due to the constant step-size, there is misadjustment in the final solution. Although no analytical expressions for misadjustments are derived in this chapter, we

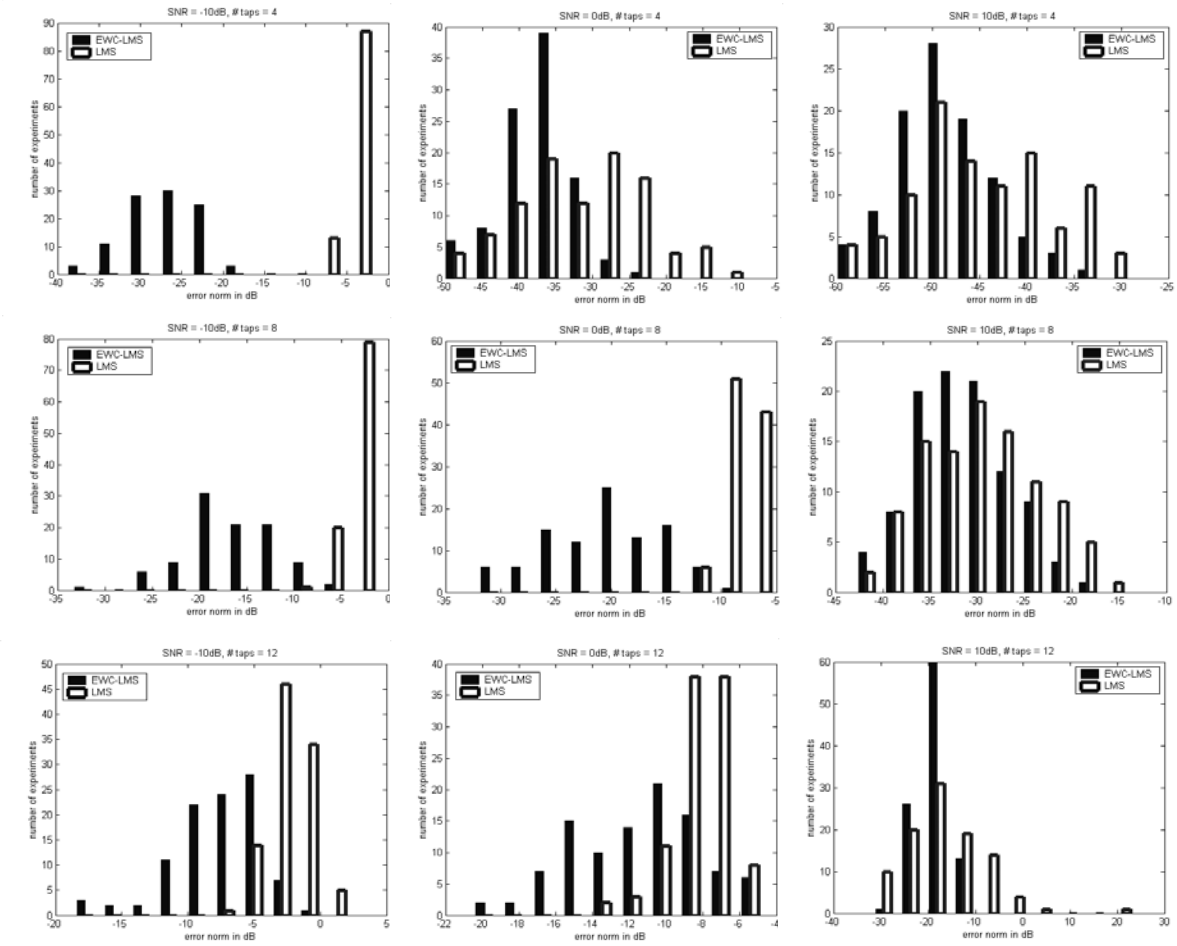


Figure 10.12. Histogram plots showing the normalized error vector norm for EWC-LMS and LMS algorithms

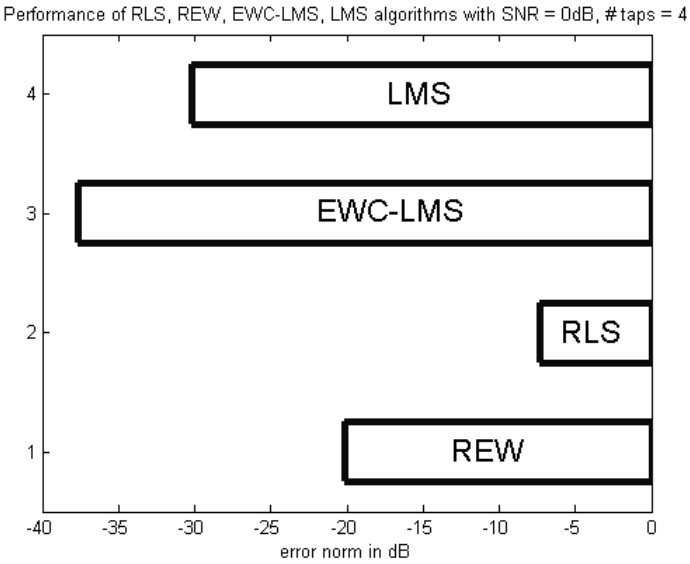
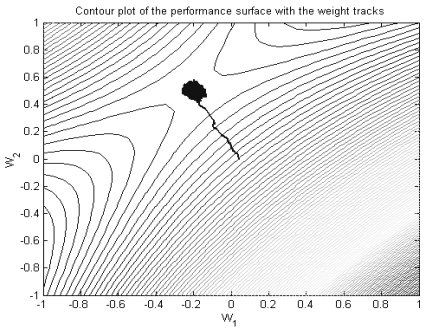


Figure 10.13. Comparison of stochastic versus recursive algorithms



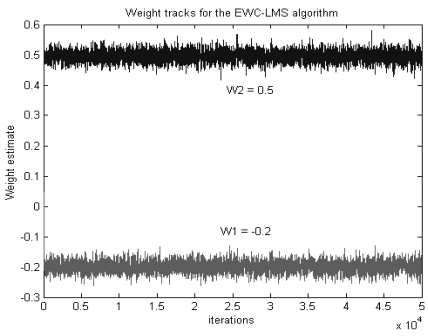


Figure 10.14. Contour plot with weight tracks Figure 10.15. Weight tracks

have done some preliminary work on estimating the misadjustment and excess-error for EWC-LMS [Rao, Erdogmus, & Principe, 2002; Erdogmus, Rao, & Principe, 2002].

In Figure 10.15, we show the individual weight tracks for the EWC-LMS algorithm. The weights converge to the vicinity of the true filter parameters, which are -0.2 and 0.5 respectively within 1000 samples. In order to see if the algorithm in (10.67) converges to the saddle point solution in a robust manner, we ran the same experiment using different initial conditions on the contours. Figure 10.16 shows a few plots of the weight tracks originating from different initial values over the contours of the performance surface. In every case, the algorithm converged to the saddle point in a stable manner. Note that the misadjustment in each case is almost the same. Finally, in order to see the effect of reducing the SNR, we repeated the experiment with 0dB SNR. Figure 10.17 (left) shows the weight tracks over the contour and we can see that there is more misadjustment now. However, we have observed that by using smaller step-sizes, the misadjustment can be controlled to be within acceptable values. Figure 10.17 (right) shows the weight tracks when the algorithm is used without the sign information for the step-size. Note that convergence is not achieved in this case which substantiates our previous argument that a fixed sign step-size will never converge to a saddle point.

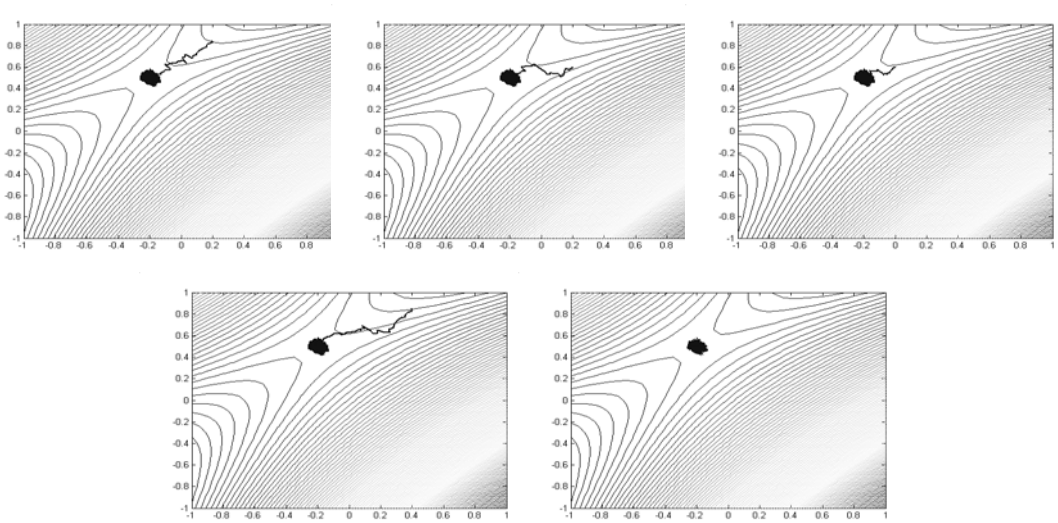


Figure 10.16. Contour plot with weight tracks for different initial values for the weights.

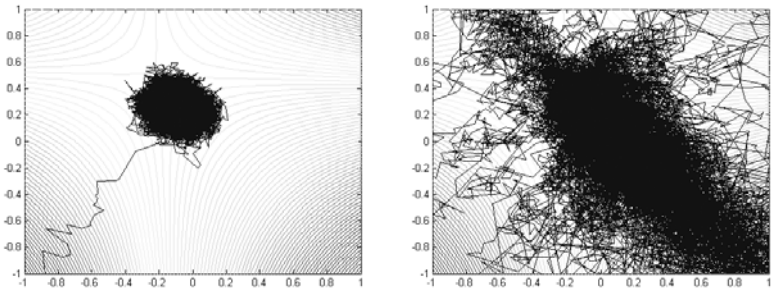


Figure 10.17. Contour plot with weight tracks for EWC-LMS algorithm with sign information (left) and without sign information (right) (0dB SNR and 2 filter taps case).

10.8. SUMMARY AND DISCUSSION

MSE has been the criterion of choice in many function approximation tasks including adaptive filter optimization. There are alternatives and enhancements to MSE that have been proposed in order to improve the robustness of learning algorithms in the presence of noisy training data. In FIR filter adaptation, noise present in the input signal is especially problematic since MSE cannot eliminate this factor. A powerful enhancement technique, total least squares, on one hand, fails to work if the noise levels in the input and output signals are not identically equal. The alternative method of subspace Wiener filtering, on the other hand, requires the noise power to be strictly smaller than the signal power to improve SNR.

We have proposed in this chapter an extension to the traditional MSE criterion in filter adaptation, which we have named the error-whitening criterion. This new cost function is inspired from the observations made on the properties of the error autocorrelation function. Specifically, we have shown that using non-zero lags of the error autocorrelation function, it is possible to obtain unbiased estimates of the model parameters even in the presence of white noise on the training data.

The new EWC criterion offers a parametric family of optimal solutions. The classical Wiener solution remains a special case corresponding to the choice $\beta = 0$, whereas total noise rejection is achieved for the special choice of $\beta = -1/2$. We have shown that the optimal solution yields an error signal uncorrelated with the predicted next value of the input vector, based on analogies with Newtonian mechanics of motion. On the other hand, the relationship with entropy through the stochastic approximation reveals a clearer understanding of the behavior of this optimal solution; the true weight vector that generated the training data marks the lags at which the error autocorrelation will become zero. We have exploited this fact to optimize the adaptive filter weights without being affected by noise.

The theoretical analysis has also been complemented by on-line algorithms that search on a sample by sample basis the optimum of the EWC. We have shown that the EWC may have a maximum, a minimum or a saddle point solution for the more interesting case of $\beta < 0$. Searching such surfaces brings difficulties for gradient descent, but search methods that use the information of the curvature work without difficulty. We have presented a recursive algorithm to find the optimum of the EWC, which is called the

REW. The REW has the same structure and complexity as the RLS algorithm. We also presented two gradient-based algorithms to search the EWC function: one that includes the curvature at the operating point, but that it has the same complexity as the REW, and is therefore uninteresting computationally. The other algorithm, which we called EWC-LMS has complexity $O(m)$ and requires the estimation of the sign of the update for the case $\beta = -0.5$. We have estimated the sign using the instantaneous estimate of the cost of the two independent functions (related to the error and its derivative). This procedure does not affect the convergence of the algorithm in the mean, but may affect the misadjustment. However this analysis is left for further research.

All in all, we have introduced a new class of Wiener type filter (the EWWF) that is able to find the optimal weights when the input data (generated by an MA process) is corrupted by additive white noise. We further develop a practical sample-by-sample fixed-point algorithm (REW) similar to RLS, and one gradient based algorithm (EWC-LMS) similar to LMS. This new class of Wiener filters represents a major advantage in many real world applications of importance in signal processing, controls and bioengineering. We studied here the simplest of this class of cost functions, where only one extra term (the first derivative) in the error vector is included. It will be important to characterize the advantages of using higher order Taylor series in the error vector in other applications such as correlated additive noise case, non-stationary data and modeling of ARMA systems. In parallel, further research on the gradient-based algorithms is also warranted. But this paper presents sufficient detail at the theoretical and algorithmic levels to enable immediate applications to real data.

Acknowledgments: This work is partially supported by the NSF grant ECS-9900394.

APPENDIX A

This appendix aims to motivate an understanding of the relationship between entropy and sample differences. In general, the parametric family describing the error pdf in supervised learning is not analytically available. In such circumstances, non-parametric approaches such as Parzen windowing [Parzen, 1967] could be employed. Given the *iid* samples {e(1),…,*e*(*N*) } of a random variable *e*, the Parzen window estimate for the underlying pdf $f_e(.)$ is obtained by

$$\hat{f}_e(x) = \frac{1}{N}\sum_{i=1}^{N} \kappa_\sigma (x - e(i)) \tag{A.1}$$

where $\kappa_\sigma(.)$ is the kernel function, which itself is a pdf, and σ is the kernel size that controls the width of each *window*. Typically, Gaussian kernels are preferred, but other kernel functions like the Cauchy density or the members of the generalized Gaussian family can be employed.

Shannon's entropy for a random variable *e* with pdf $f_e(.)$ is defined as [Shannon & Weaver, 1964]

$$H(e) = -\int_{-\infty}^{\infty} f_e(x) \log f_e(x) dx = -E_e[f_e(e)] \tag{A.2}$$

Given *iid* samples, this entropy could be estimated using [Erdogmus, 2002]

$$\hat{H}(e) = -\frac{1}{N}\sum_{j=1}^{N} \log\left(\frac{1}{N}\sum_{i=1}^{N} \kappa_\sigma(e(j) - e(i))\right) \tag{A.3}$$

This estimator uses the sample mean approximation for the expected value and the Parzen window estimator for the pdf. Viola proposed a similar entropy estimator, in which he suggested dividing the samples into two subsets: one for estimating the pdf, the other for evaluating the sample mean [Viola, Schraudolph, & Sejnowski, 1995]. In order to approximate a stochastic entropy estimator, we approximate the expectation by evaluating the argument at the most recent sample, e_k. In order to estimate the pdf, we use the L previous samples. The stochastic entropy estimator then becomes

$$\overline{H}(e) = -\log\left(\frac{1}{L}\sum_{i=1}^{L} \kappa_\sigma(e(k) - e(i))\right) \tag{A.4}$$

For supervised training of an ADALINE (or an FIR filter), with weight vector $w \in \Re^n$, given the input (vector)-desired training sequence $(\mathbf{x}(n),d(n))$, where $\mathbf{x}(n) \in \Re^m$ and $d(n) \in \Re$, the instantaneous error is given by $e(n) = d(n) - \mathbf{w}^T(n)\mathbf{x}(n)$. The stochastic gradient of the error entropy with respect to the weights becomes

$$\frac{\partial \overline{H}(X)}{\partial \mathbf{w}} = -\left(\sum_{i=1}^{L} \kappa'_\sigma(e(n) - e(n-i))\big(\mathbf{x}(n) - \mathbf{x}(n-i)\big)\right) \Bigg/ \sum_{i=1}^{L} \kappa_\sigma(e(n) - e(n-i)) \tag{A.5}$$

where $e(n-i) = d(n-i) - \mathbf{w}^T(n)\mathbf{x}(n-i)$ is also evaluated using the same weight vector as $e(n)$ [Erdogmus, 2002]. For the specific choice of a single error sample $e(k\text{-}L)$ for pdf estimation and a Gaussian kernel function, (A.5) reduces to

$$\frac{\partial \overline{H}(X)}{\partial \mathbf{w}} = -(e(n) - e(n-L))\big(\mathbf{x}(n) - \mathbf{x}(n-L)\big)/\sigma^2 \tag{A.6}$$

We easily notice that the expression in (A.6) is also a stochastic gradient for the cost function $J = E[(e(n) - e(n-L))^2]/(2\sigma^2)$.

APPENDIX B

Consider the correlation matrices **R**, **S**, **P**, and **Q** estimated from noisy data. For **R**, we write

$$\begin{aligned}\mathbf{R} &= E[\mathbf{x}(n)\mathbf{x}^T(n)] = E[(\tilde{\mathbf{x}}(n)+\mathbf{v}(n))(\tilde{\mathbf{x}}(n)+\mathbf{v}(n))^T] \\ &= E[\tilde{\mathbf{x}}(n)\tilde{\mathbf{x}}^T(n)+\tilde{\mathbf{x}}(n)\mathbf{v}^T(n)+\mathbf{v}(n)\mathbf{x}^T(n)+\mathbf{v}(n)\mathbf{v}^T(n)] \\ &= E[\tilde{\mathbf{x}}(n)\tilde{\mathbf{x}}^T(n)+\mathbf{v}(n)\mathbf{v}^T(n)] = \tilde{\mathbf{R}}+\mathbf{V}\end{aligned} \tag{B.1}$$

For **S**, we obtain

$$\begin{aligned}\mathbf{S} &= E[\mathbf{x}(n)\mathbf{x}^T(n)+\mathbf{x}(n)\mathbf{x}^T(n)-\mathbf{x}(n)\mathbf{x}^T(n-L)-\mathbf{x}(n-L)\mathbf{x}^T(n)] \\ &= 2\mathbf{R}-E[(\tilde{\mathbf{x}}(n)+\mathbf{v}(n))(\tilde{\mathbf{x}}(n-L)+\mathbf{v}(n-L))^T+(\tilde{\mathbf{x}}(n-L)+\mathbf{v}(n-L))(\tilde{\mathbf{x}}(n)+\mathbf{v}(n))^T] \\ &= 2(\tilde{\mathbf{R}}+\mathbf{V})-E\begin{bmatrix}\tilde{\mathbf{x}}(n)\tilde{\mathbf{x}}^T(n-L)+\tilde{\mathbf{x}}(n)\mathbf{v}^T(n-L)+\mathbf{v}(n)\tilde{\mathbf{x}}^T(n-L)+\mathbf{v}(n)\mathbf{v}^T(n-L) \\ +\tilde{\mathbf{x}}(n-L)\tilde{\mathbf{x}}^T(n)+\tilde{\mathbf{x}}(n-L)\mathbf{v}^T(n)+\mathbf{v}(n-L)\tilde{\mathbf{x}}^T(n)+\mathbf{v}(n-L)\mathbf{v}^T(n)\end{bmatrix} \\ &= 2(\tilde{\mathbf{R}}+\mathbf{V})-E[\tilde{\mathbf{x}}(n)\tilde{\mathbf{x}}^T(n-L)+\tilde{\mathbf{x}}(n-L)\tilde{\mathbf{x}}^T(n)]-E[\mathbf{v}(n)\mathbf{v}^T(n-L)+\mathbf{v}(n-L)\mathbf{v}^T(n)] \\ &= 2(\tilde{\mathbf{R}}+\mathbf{V})-\tilde{\mathbf{R}}_L-\mathbf{V}_L\end{aligned} \tag{B.2}$$

Similarly, for **P** and **Q** we get

$$\begin{aligned}\mathbf{P} &= E[\mathbf{x}(n)d(n)] = E[(\tilde{\mathbf{x}}(n)+\mathbf{v}(n))(\tilde{d}(n)+u(n))] \\ &= E[\tilde{\mathbf{x}}(n)\tilde{d}(n)+\tilde{\mathbf{x}}(n)u(n)+\mathbf{v}(n)\tilde{d}(n)+\mathbf{v}(n)u(n)] \\ &= E[\tilde{\mathbf{x}}(n)\tilde{d}(n)] = \tilde{\mathbf{P}}\end{aligned} \tag{B.3}$$

$$\begin{aligned}\mathbf{Q} &= E[(\mathbf{x}(n)-\mathbf{x}(n-L))(d(n)-d(n-L))] \\ &= E[\mathbf{x}(n)d(n)-\mathbf{x}(n)d(n-L)-\mathbf{x}(n-L)d(n)+\mathbf{x}(n-L)d(n-L)] \\ &= 2\mathbf{P}-E[\mathbf{x}(n)d(n-L)+\mathbf{x}(n-L)d(n)] \\ &= 2\tilde{\mathbf{P}}-E[(\tilde{\mathbf{x}}(n)+\mathbf{v}(n))(\tilde{d}(n-L)+u(n-L))+(\tilde{\mathbf{x}}(n-L)+\mathbf{v}(n-L))(\tilde{d}(n)+u(n))] \\ &= 2\tilde{\mathbf{P}}-E[\tilde{\mathbf{x}}(n)\tilde{d}(n-L)+\tilde{\mathbf{x}}(n-L)\tilde{d}(n)]-E\begin{bmatrix}\tilde{\mathbf{x}}(n)u(n-L)+\mathbf{v}(n)\tilde{d}(n-L) \\ +\mathbf{v}(n)u(n-L)+\tilde{\mathbf{x}}(n-L)u(n) \\ +\mathbf{v}(n-L)\tilde{d}(n)+\mathbf{v}(n-L)u(n)\end{bmatrix} \\ &= 2\tilde{\mathbf{P}}-\tilde{\mathbf{P}}_L\end{aligned} \tag{B.4}$$

APPENDIX C

Recall that the optimal solution of EWC satisfies (10.9), which is equivalently

$$E[(1+2\beta)e(n)\mathbf{x}(n)-\beta(e(n)\mathbf{x}(n-L)+e(n)\mathbf{x}(n+L))]=0 \tag{C.1}$$

Rearranging the terms in (C.1), we obtain

$$E[e(n)(\mathbf{x}(n)-\beta(\mathbf{x}(n+L)-2\mathbf{x}(n)-\mathbf{x}(n-L)))]=0 \tag{C.2}$$

Notice that the combination of x-values that multiply β form an estimate of the *acceleration* of the input vector $\mathbf{x}(n)$. Specifically for $\beta = -1/2$, the term that multiplies $e(n)$ becomes a single-step prediction for the input vector $\mathbf{x}(n)$ (assuming zero velocity and constant acceleration), according to Newtonian mechanics. Thus, the optimal solution of the EWC criterion tries decorrelating the error signal from the predicted next input vector.

REFERENCES

[1] H. Akaike, "A New Look at the Statistical Model Identification," *IEEE Trans. Automatic Control*, vol. 19, pp. 716-723, 1974.

[2] C. Beck, F. Schlogl, *Thermodynamics of Chaotic Systems*, Cambridge University Press, Cambridge, 1993.

[3] J. Beirlant, M.C.A. Zuijlen, "The Empirical Distribution Function and Strong Laws for Functions of Order Statistics of Uniform Spacings," *Journal of Multivariate Analysis*, vol. 16, pp. 300-317, 1985.

[4] A. Benveniste, M. Metivier, P. Priouret, "*Adaptive Algorithms and Stochastic Approximations*". Springer-Verlag, 1990.

[5] P.J. Bickel, L. Breiman, "Sums of Functions of Nearest Neighbor Distances, Moment Bounds, Limit Theorems and a Goodness-of-fit Test," *Annals of Statistics*, vol. 11, pp. 185-214, 1983.

[6] C. Bishop, *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, 1995.

[7] J.A. Cadzow, "Total Least Squares, Matrix Enhancement, and Signal Processing," *Digital Signal Processing*, vol. 4, pp. 21-39, 1994.

[8] M. Chansarkar, U.B. Desai, "A Robust Recursive Least Squares Algorithm", *IEEE Trans. Signal Processing*, vol. 45, no. 7, July 1997.

[9] B. de Moor, "Total Least Squares for Affinely Structured Matrices and the Noisy Realization Problem," *IEEE Trans. Signal Processing*, vol. 42, pp. 3104-3113, 1994.

[10] S.C. Douglas, W. Pan, "Exact Expectation Analysis of the LMS Adaptive Filter", *IEEE. Trans. Signal Processing*. Vol. 43, pp-2863-2871, 1995.

[11] S. C. Douglas, "Analysis of an Anti-Hebbian Adaptive FIR Filtering Algorithm". *IEEE Trans. Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 43, no. 11, November 1996.

[12] D. Erdogmus, "*Information Theoretic Learning: Renyi's Entropy and its Applications to Adaptive System Training*," PhD Dissertation, University of Florida, Gainesville, FL, 2002.

[13] D. Erdogmus, J.C. Principe, "An On-Line Adaptation Algorithm for Adaptive System Training with Minimum Error Entropy: Stochastic Information Gradient," *Proceedings of ICA'01*, pp. 7-12, San Diego, CA, 2001.

[14] D. Erdogmus, J.C. Principe, "Generalized Information Potential Criterion for Adaptive System Training", to appear in *IEEE Trans. Neural Networks*, Sept. 2002.

[15] D. Erdogmus, J.C. Principe, K.E. Hild II, "Do Hebbian Synapses Estimate Entropy?," accepted to *NNSP'02*, 2002.

[16] D. Erdogmus, Y.N. Rao, J.C. Principe, "An Error Whitening Criterion for Adaptive Filtering – Part I: The Theory". Submitted to *IEEE Trans. Signal Processing*.
[17] B. Farhang-Boroujeny, *Adaptive Filters: Theory and Applications*, Wiley, New York, 1998.
[18] D.Z Feng, Z. Bao, L.C. Jiao, "Total Least Mean Squares Algorithm," *IEEE Trans. Signal Processing*, vol. 46, no. 8, pp. 2122-2130, 1998.
[19] K. Gao, M.O. Ahmad, M.N.S. Swamy, "A Constrained Anti-Hebbian Learning Algorithm for Total Least Squares Estimation with Applications to Adaptive FIR and IIR Filtering," *IEEE Trans. Circuits and Systems Part 2*, vol. 41, no. 11, pp. 718-729, 1994.
[20] G.H. Golub, C.F. van Loan, "An Analysis of the Total Least Squares Problem," *SIAM Journal of Numerical Analysis*, vol. 17, no. 4, pp. 883-893, 1979.
[21] G.H. Golub, C.F. van Loan, *Matrix Computations*, Baltimore, MD, Johns Hopkins Univ. Press, 1989.
[22] P. Hall, "Limit Theorems for Sums of General Functions of *m*-Spacings," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 96 pp. 517-532, 1984.
[23] S. Haykin, "*Neural Networks: A comprehensive Foundation*", Macmillan, New York, 1994.
[24] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, Upper Saddle River, New Jersey, 1996.
[25] L.F. Kozachenko, N.N. Leonenko, "Sample Estimate of Entropy of a Random Vector," *Problems of Information Transmission*, vol. 23, pp. 95-101, 1987.
[26] H.J. Kushner, D.S. Clark, "*Stochastic Approximation Methods for Constrained and Unconstrained Systems*", New York: Springer-Verlag, 1978.
[27] P. Lemmerling, "*Structured Total Least Squares: Analysis, Algorithms, and Applications*," PhD Dissertation, Katholeike University, Leuven, Belgium 1999.
[28] L. Ljung, "Analysis of recursive stochastic algorithms". *IEEE Transactions on Automatic Control*, vol. AC-22, pp. 551-575, 1977.
[29] M. Mueller, "Least-Squares Algorithms for Adaptive Equalizers", *Bell Systems Technical Journal*, vol. 60, pp. 1905-1925, 1981.
[30] E. Parzen, "On Estimation of a Probability Density Function and Mode", in *Time Series Analysis Papers*, Holden-Day, Inc., San Diego, California, 1967.
[31] J.C. Principe, N. Euliano, C. Lefebvre, *Neural and Adaptive Systems: Fundamentals Through Simulations*, Wiley, New York, 1999.
[32] Y.N. Rao, *Algorithms for Eigendecomposition and Time Series Segmentation*, MS Thesis, University of Florida, Gainesville, FL, 2000.
[33] Y.N. Rao, D. Erdogmus, J.C. Principe, "An Error Whitening Criterion for Adaptive Filtering – Part II: Algorithms". Submitted to *IEEE Trans. Signal Processing*.
[34] Y.N. Rao, J.C. Principe, "Efficient Total Least Squares Method for System Modeling Using Minor Component Analysis," accepted to *NNSP'02*, 2002.
[35] P.A. Regalia, "*Adaptive IIR Filtering in Signal Processing and Control*". Marcel Dekker, 1995.
[36] M. Reuter, K. Quirk, J. Zeidler, L. Milstein, "Non-Linear Effects in LMS Adaptive Filters", *Proceedings of IEEE 2000 AS-SPCC*, October, 2000.

[37] J. Rissanen, *Stochastic Complexity in Statistical Inquiry*, World Scientific, London, 1989.
[38] C.E. Shannon, W. Weaver, *The Mathematical Theory of Communication*, University of Illinois Press, Urbana, 1964.
[39] H.C. So, "Modified LMS Algorithm for Unbiased Impulse Response Estimation in Nonstationary Noise," *IEE Electronics Letters*, vol. 35, no. 10, pp. 791-792, 1999.
[40] F.P. Tarasenko, "On the Evaluation of an Unknown Probability Density Function, the Direct Estimation of the Entropy from Independent Observations of a Continuous Random Variable, and the Distribution-Free Entropy Test of Goodness-of-fit," *Proceedings of IEEE*, vol. 56, pp. 2052-2053, 1968.
[41] A.B. Tsybakov, E.C. van der Meulen, "Root-n Consistent Estimators of Entropy for Densities with Unbounded Support," *Scandinavian Journal of Statistics*, vol. 23, pp. 75-83, 1994.
[42] P. Viola, N. Schraudolph, T. Sejnowski, "Empirical Entropy Manipulation for Real-World Problems*", Proceedings of NIPS'95*, pp. 851-857, 1995.
[43] A. Yeredor, "The Extended Least Squares Criterion: Minimization Algorithms and Applications," *IEEE Trans. Signal Processing*, vol. 49, no. 1, pp. 74-86, 2000.