Chapter 3
The Least-Mean-Square (LMS) Algorithm

3.1 Introduction

The least-mean-square (LMS) is a search algorithm in which a simplification of the gradient vector computation is made possible by appropriately modifying the objective function [1,2]. The LMS algorithm, as well as others related to it, is widely used in various applications of adaptive filtering due to its computational simplicity [3–7]. The convergence characteristics of the LMS algorithm are examined in order to establish a range for the convergence factor that will guarantee stability. The convergence speed of the LMS is shown to be dependent on the eigenvalue spread of the input signal correlation matrix [2–6]. In this chapter, several properties of the LMS algorithm are discussed including the misadjustment in stationary and nonstationary environments [2–9] and tracking performance [10–12]. The analysis results are verified by a large number of simulation examples. Chapter 15, Sect. 15.1, complements this chapter by analyzing the finite-wordlength effects in LMS algorithms.

The LMS algorithm is by far the most widely used algorithm in adaptive filtering for several reasons. The main features that attracted the use of the LMS algorithm are low computational complexity, proof of convergence in stationary environment, unbiased convergence in the mean to the Wiener solution, and stable behavior when implemented with finite-precision arithmetic. The convergence analysis of the LMS presented here utilizes the independence assumption.

3.2 The LMS Algorithm

In Chap. 2 we derived the optimal solution for the parameters of the adaptive filter implemented through a linear combiner, which corresponds to the case of multiple input signals. This solution leads to the minimum mean-square error in estimating

P.S.R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*,
DOI 10.1007/978-1-4614-4106-9_3, © Springer Science+Business Media New York 2013

the reference signal $d(k)$. The optimal (Wiener) solution is given by

$$\mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p} \tag{3.1}$$

where $\mathbf{R} = E[\mathbf{x}(k)\mathbf{x}^T(k)]$ and $\mathbf{p} = E[d(k)\mathbf{x}(k)]$, assuming that $d(k)$ and $\mathbf{x}(k)$ are jointly WSS.

If good estimates of matrix $\mathbf{R}$, denoted by $\hat{\mathbf{R}}(k)$, and of vector $\mathbf{p}$, denoted by $\hat{\mathbf{p}}(k)$, are available, a steepest-descent-based algorithm can be used to search the Wiener solution of (3.1) as follows:

$$\begin{aligned}\mathbf{w}(k+1) &= \mathbf{w}(k) - \mu\hat{\mathbf{g}}_{\mathbf{w}}(k)\\ &= \mathbf{w}(k) + 2\mu(\hat{\mathbf{p}}(k) - \hat{\mathbf{R}}(k)\mathbf{w}(k))\end{aligned} \tag{3.2}$$

for $k = 0, 1, 2, \ldots$, where $\hat{\mathbf{g}}_{\mathbf{w}}(k)$ represents an estimate of the gradient vector of the objective function with respect to the filter coefficients.

One possible solution is to estimate the gradient vector by employing instantaneous estimates for $\mathbf{R}$ and $\mathbf{p}$ as follows:

$$\begin{aligned}\hat{\mathbf{R}}(k) &= \mathbf{x}(k)\mathbf{x}^T(k)\\ \hat{\mathbf{p}}(k) &= d(k)\mathbf{x}(k)\end{aligned} \tag{3.3}$$

The resulting gradient estimate is given by

$$\begin{aligned}\hat{\mathbf{g}}_{\mathbf{w}}(k) &= -2d(k)\mathbf{x}(k) + 2\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{w}(k)\\ &= 2\mathbf{x}(k)(-d(k) + \mathbf{x}^T(k)\mathbf{w}(k))\\ &= -2e(k)\mathbf{x}(k)\end{aligned} \tag{3.4}$$

Note that if the objective function is replaced by the instantaneous square error $e^2(k)$, instead of the MSE, the above gradient estimate represents the true gradient vector since

$$\begin{aligned}\frac{\partial e^2(k)}{\partial \mathbf{w}} &= \left[2e(k)\frac{\partial e(k)}{\partial w_0(k)}\; 2e(k)\frac{\partial e(k)}{\partial w_1(k)}\; \ldots\; 2e(k)\frac{\partial e(k)}{\partial w_N(k)}\right]^T\\ &= -2e(k)\mathbf{x}(k)\\ &= \hat{\mathbf{g}}_{\mathbf{w}}(k)\end{aligned} \tag{3.5}$$

The resulting gradient-based algorithm is known[1] as the least-mean-square (LMS) algorithm, whose updating equation is

$$\mathbf{w}(k+1) = \mathbf{w}(k) + 2\mu e(k)\mathbf{x}(k) \tag{3.6}$$

where the convergence factor μ should be chosen in a range to guarantee convergence.

[1] Because it minimizes the mean of the squared error.

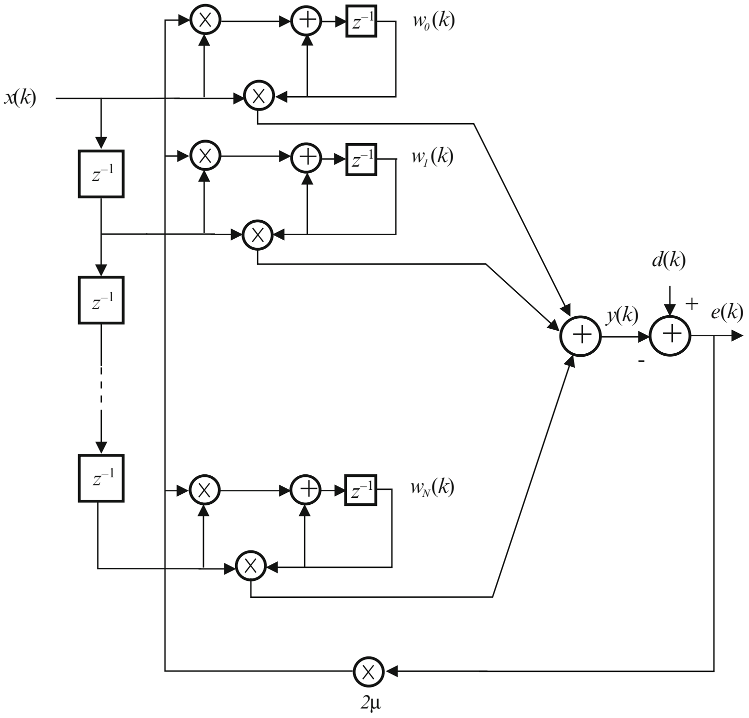


Fig. 3.1 LMS adaptive FIR filter

Algorithm 3.1 LMS algorithm

```
Initialization
 $\mathbf{x}(0) = \mathbf{w}(0) = [0\,0\,\ldots\,0]^T$
Do for $k \geq 0$
 $e(k) = d(k) - \mathbf{x}^T(k)\mathbf{w}(k)$
 $\mathbf{w}(k+1) = \mathbf{w}(k) + 2\mu e(k)\mathbf{x}(k)$
```

Figure 3.1 depicts the realization of the LMS algorithm for a delay line input $\mathbf{x}(k)$. Typically, one iteration of the LMS requires $N+2$ multiplications for the filter coefficient updating and $N+1$ multiplications for the error generation. The detailed description of the LMS algorithm is shown in the table denoted as Algorithm 3.1.

It should be noted that the initialization is not necessarily performed as described in Algorithm 3.1, where the coefficients of the adaptive filter were initialized with zeros. For example, if a rough idea of the optimal coefficient value is known, these values could be used to form $\mathbf{w}(0)$ leading to a reduction in the number of iterations required to reach the neighborhood of $\mathbf{w}_o$.

3.3 Some Properties of the LMS Algorithm

In this section, the main properties related to the convergence behavior of the LMS algorithm in a stationary environment are described. The information contained here is essential to understand the influence of the convergence factor μ in various convergence aspects of the LMS algorithm.

3.3.1 Gradient Behavior

As shown in Chap. 2, see (2.91), the ideal gradient direction required to perform a search on the MSE surface for the optimum coefficient vector solution is

$$\begin{aligned} \mathbf{g_w}(k) &= 2\left\{E\left[\mathbf{x}(k)\mathbf{x}^T(k)\right]\mathbf{w}(k) - E\left[d(k)\mathbf{x}(k)\right]\right\} \\ &= 2[\mathbf{R}\mathbf{w}(k) - \mathbf{p}] \end{aligned} \tag{3.7}$$

In the LMS algorithm, instantaneous estimates of $\mathbf{R}$ and $\mathbf{p}$ are used to determine the search direction, i.e.,

$$\hat{\mathbf{g}}_{\mathbf{w}}(k) = 2\left[\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{w}(k) - d(k)\mathbf{x}(k)\right] \tag{3.8}$$

As can be expected, the direction determined by (3.8) is quite different from that of (3.7). Therefore, by using the more computationally attractive gradient direction of the LMS algorithm, the convergence behavior is not the same as that of the steepest-descent algorithm.

On average, it can be said that the LMS gradient direction has the tendency to approach the ideal gradient direction since for a fixed coefficient vector $\mathbf{w}$

$$\begin{aligned} E[\hat{\mathbf{g}}_{\mathbf{w}}(k)] &= 2\{E\left[\mathbf{x}(k)\mathbf{x}^T(k)\right]\mathbf{w} - E\left[d(k)\mathbf{x}(k)\right]\} \\ &= \mathbf{g_w} \end{aligned} \tag{3.9}$$

hence, vector $\hat{\mathbf{g}}_{\mathbf{w}}(k)$ can be interpreted as an unbiased instantaneous estimate of $\mathbf{g_w}$. In an ergodic environment, if, for a fixed $\mathbf{w}$ vector, $\hat{\mathbf{g}}_{\mathbf{w}}(k)$ is calculated for a large number of inputs and reference signals, the average direction tends to $\mathbf{g_w}$, i.e.,

$$\lim_{M\to\infty} \frac{1}{M} \sum_{i=1}^{M} \hat{\mathbf{g}}_{\mathbf{w}}(k+i) \to \mathbf{g_w} \tag{3.10}$$

3.3.2 Convergence Behavior of the Coefficient Vector

Assume that an unknown FIR filter with coefficient vector given by $\mathbf{w}_o$ is being identified by an adaptive FIR filter of the same order, employing the LMS algorithm. Measurement white noise $n(k)$ with zero mean and variance σ_n^2 is added to the output of the unknown system.

The error in the adaptive-filter coefficients as related to the ideal coefficient vector $\mathbf{w}_o$, in each iteration, is described by the $N + 1$-length vector

$$\Delta\mathbf{w}(k) = \mathbf{w}(k) - \mathbf{w}_o \tag{3.11}$$

With this definition, the LMS algorithm can alternatively be described by

$$\begin{aligned}\Delta\mathbf{w}(k+1) &= \Delta\mathbf{w}(k) + 2\mu e(k)\mathbf{x}(k)\\ &= \Delta\mathbf{w}(k) + 2\mu\mathbf{x}(k)\left[\mathbf{x}^T(k)\mathbf{w}_o + n(k) - \mathbf{x}^T(k)\mathbf{w}(k)\right]\\ &= \Delta\mathbf{w}(k) + 2\mu\mathbf{x}(k)\left[e_o(k) - \mathbf{x}^T(k)\Delta\mathbf{w}(k)\right]\\ &= \left[\mathbf{I} - 2\mu\mathbf{x}(k)\mathbf{x}^T(k)\right]\Delta\mathbf{w}(k) + 2\mu e_o(k)\mathbf{x}(k)\end{aligned} \tag{3.12}$$

where $e_o(k)$ is the optimum output error given by

$$\begin{aligned}e_o(k) &= d(k) - \mathbf{w}_o^T\mathbf{x}(k)\\ &= \mathbf{w}_o^T\mathbf{x}(k) + n(k) - \mathbf{w}_o^T\mathbf{x}(k)\\ &= n(k)\end{aligned} \tag{3.13}$$

The expected error in the coefficient vector is then given by

$$E[\Delta\mathbf{w}(k+1)] = E\{[\mathbf{I} - 2\mu\mathbf{x}(k)\mathbf{x}^T(k)]\Delta\mathbf{w}(k)\} + 2\mu E[e_o(k)\mathbf{x}(k)] \tag{3.14}$$

If it is assumed that the elements of $\mathbf{x}(k)$ are statistically independent of the elements of $\Delta\mathbf{w}(k)$ and $e_o(k)$, (3.14) can be simplified as follows:

$$\begin{aligned}E[\Delta\mathbf{w}(k+1)] &= \{\mathbf{I} - 2\mu E[\mathbf{x}(k)\mathbf{x}^T(k)]\}E[\Delta\mathbf{w}(k)]\\ &= (\mathbf{I} - 2\mu\mathbf{R})E[\Delta\mathbf{w}(k)]\end{aligned} \tag{3.15}$$

The first assumption is justified if we assume that the deviation in the parameters is dependent on previous input signal vectors only, whereas in the second assumption we also considered that the error signal at the optimal solution is orthogonal to the elements of the input signal vector. The above expression leads to

$$E[\Delta\mathbf{w}(k+1)] = (\mathbf{I} - 2\mu\mathbf{R})^{k+1}E[\Delta\mathbf{w}(0)] \tag{3.16}$$

Equation (3.15) premultiplied by $\mathbf{Q}^T$, where $\mathbf{Q}$ is the unitary matrix that diagonalizes $\mathbf{R}$ through a similarity transformation, yields

$$\begin{aligned} E\left[\mathbf{Q}^T\Delta\mathbf{w}(k+1)\right] &= (\mathbf{I}-2\mu\mathbf{Q}^T\mathbf{R}\mathbf{Q})E\left[\mathbf{Q}^T\Delta\mathbf{w}(k)\right] \\ &= E\left[\Delta\mathbf{w}'(k+1)\right] \\ &= (\mathbf{I}-2\mu\boldsymbol{\Lambda})E\left[\Delta\mathbf{w}'(k)\right] \\ &= \begin{bmatrix} 1-2\mu\lambda_0 & 0 & \cdots & 0 \\ 0 & 1-2\mu\lambda_1 & & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & & 1-2\mu\lambda_N \end{bmatrix} E\left[\Delta\mathbf{w}'(k)\right] \end{aligned} \tag{3.17}$$

where $\Delta\mathbf{w}'(k+1) = \mathbf{Q}^T\Delta\mathbf{w}(k+1)$ is the rotated-coefficient error vector. The applied rotation yielded an equation where the driving matrix is diagonal, making it easier to analyze the equation's dynamic behavior. Alternatively, the above relation can be expressed as

$$\begin{aligned} E\left[\Delta\mathbf{w}'(k+1)\right] &= (\mathbf{I}-2\mu\boldsymbol{\Lambda})^{k+1}E\left[\Delta\mathbf{w}'(0)\right] \\ &= \begin{bmatrix} (1-2\mu\lambda_0)^{k+1} & 0 & \cdots & 0 \\ 0 & (1-2\mu\lambda_1)^{k+1} & & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & & (1-2\mu\lambda_N)^{k+1} \end{bmatrix} E\left[\Delta\mathbf{w}'(0)\right] \end{aligned} \tag{3.18}$$

This equation shows that in order to guarantee convergence of the coefficients in the mean, the convergence factor of the LMS algorithm must be chosen in the range

$$0 < \mu < \frac{1}{\lambda_{\max}} \tag{3.19}$$

where $\lambda_{\max}$ is the largest eigenvalue of $\mathbf{R}$. Values of μ in this range guarantees that all elements of the diagonal matrix in (3.18) tend to zero as $k \rightarrow \infty$, since $-1 < (1-2\mu\lambda_i) < 1$, for $i = 0, 1, \ldots, N$. As a result $E[\Delta\mathbf{w}'(k+1)]$ tends to zero for large k.

The choice of μ as above explained ensures that the mean value of the coefficient vector approaches the optimum coefficient vector $\mathbf{w}_o$. It should be mentioned that if the matrix $\mathbf{R}$ has a large eigenvalue spread, it is advisable to choose a value for μ much smaller than the upper bound. As a result, the convergence speed of the coefficients will be primarily dependent on the value of the smallest eigenvalue, responsible for the slowest mode in (3.18).

The key assumption for the above analysis is the so-called independence theory [4], which considers all vectors $\mathbf{x}(i)$, for $i = 0, 1, \ldots, k$, statistically independent. This assumption allowed us to consider $\Delta\mathbf{w}(k)$ independent of $\mathbf{x}(k)\mathbf{x}^T(k)$ in (3.14). Such an assumption, despite not being rigorously valid especially when $\mathbf{x}(k)$ consists of the elements of a delay line, leads to theoretical results that are in good agreement with the experimental results.

3.3.3 Coefficient-Error-Vector Covariance Matrix

In this subsection, we derive the expressions for the second-order statistics of the errors in the adaptive-filter coefficients. Since for large k the mean value of $\Delta\mathbf{w}(k)$ is zero, the covariance of the coefficient-error vector is defined as

$$\text{cov}[\Delta\mathbf{w}(k)] = E[\Delta\mathbf{w}(k)\Delta\mathbf{w}^T(k)] = E\left\{[\mathbf{w}(k) - \mathbf{w}_o][\mathbf{w}(k) - \mathbf{w}_o]^T\right\} \tag{3.20}$$

By replacing (3.12) in (3.20) it follows that

$$\begin{aligned}\text{cov}[\Delta\mathbf{w}(k+1)] &= E\Big\{\left[\mathbf{I} - 2\mu\mathbf{x}(k)\mathbf{x}^T(k)\right]\Delta\mathbf{w}(k)\Delta\mathbf{w}^T(k)\left[\mathbf{I} - 2\mu\mathbf{x}(k)\mathbf{x}^T(k)\right]^T \\ &\quad + [\mathbf{I} - 2\mu\mathbf{x}(k)\mathbf{x}^T(k)]\Delta\mathbf{w}(k)2\mu e_o(k)\mathbf{x}^T(k) \\ &\quad + 2\mu e_o(k)\mathbf{x}(k)\Delta\mathbf{w}^T(k)[\mathbf{I} - 2\mu\mathbf{x}(k)\mathbf{x}^T(k)]^T \\ &\quad + 4\mu^2 e_o^2(k)\mathbf{x}(k)\mathbf{x}^T(k)\Big\}\end{aligned} \tag{3.21}$$

By considering $e_o(k)$ independent of $\Delta\mathbf{w}(k)$ and orthogonal to $\mathbf{x}(k)$, the second and third terms on the right-hand side of the above equation can be eliminated. The details of this simplification can be carried out by describing each element of the eliminated matrices explicitly. In this case,

$$\begin{aligned}\text{cov}[\Delta\mathbf{w}(k+1)] = \text{cov}[\Delta\mathbf{w}(k)] + E\big[&-2\mu\mathbf{x}(k)\mathbf{x}^T(k)\Delta\mathbf{w}(k)\Delta\mathbf{w}^T(k) \\ &- 2\mu\Delta\mathbf{w}(k)\Delta\mathbf{w}^T(k)\mathbf{x}(k)\mathbf{x}^T(k) \\ &+ 4\mu^2\mathbf{x}(k)\mathbf{x}^T(k)\Delta\mathbf{w}(k)\Delta\mathbf{w}^T(k)\mathbf{x}(k)\mathbf{x}^T(k) \\ &+ 4\mu^2 e_o^2(k)\mathbf{x}(k)\mathbf{x}^T(k)\big]\end{aligned} \tag{3.22}$$

In addition, assuming that $\Delta\mathbf{w}(k)$ and $\mathbf{x}(k)$ are independent, (3.22) can be rewritten as

$$\begin{aligned}\text{cov}[\Delta\mathbf{w}(k+1)] = \text{cov}[\Delta\mathbf{w}(k)] &- 2\mu E[\mathbf{x}(k)\mathbf{x}^T(k)]E[\Delta\mathbf{w}(k)\Delta\mathbf{w}^T(k)] \\ &- 2\mu E[\Delta\mathbf{w}(k)\Delta\mathbf{w}^T(k)]E[\mathbf{x}(k)\mathbf{x}^T(k)] \\ &+ 4\mu^2 E\left\{\mathbf{x}(k)\mathbf{x}^T(k)E[\Delta\mathbf{w}(k)\Delta\mathbf{w}^T(k)]\mathbf{x}(k)\mathbf{x}^T(k)\right\} \\ &+ 4\mu^2 E[e_o^2(k)]E[\mathbf{x}(k)\mathbf{x}^T(k)]\end{aligned}$$

$$\begin{aligned}&= \text{cov}[\Delta\mathbf{w}(k)] - 2\mu\mathbf{R}\,\text{cov}[\Delta\mathbf{w}(k)]\\ &\quad - 2\mu\,\text{cov}[\Delta\mathbf{w}(k)]\mathbf{R} + 4\mu^2\mathbf{A} + 4\mu^2\sigma_n^2\mathbf{R}\end{aligned} \tag{3.23}$$

The calculation of $\mathbf{A} = E\left\{\mathbf{x}(k)\mathbf{x}^T(k)E[\Delta\mathbf{w}(k)\Delta\mathbf{w}^T(k)]\mathbf{x}(k)\mathbf{x}^T(k)\right\}$ involves fourth-order moments and the result can be obtained by expanding the matrix inside the operation $E[\cdot]$ as described in [4] and [13] for jointly Gaussian input signal samples. The result is

$$\mathbf{A} = 2\mathbf{R}\,\text{cov}[\Delta\mathbf{w}(k)]\,\mathbf{R} + \mathbf{R}\,\text{tr}\{\mathbf{R}\,\text{cov}[\Delta\mathbf{w}(k)]\} \tag{3.24}$$

where tr$[\cdot]$ denotes trace of $[\cdot]$. Equation (3.23) is needed to calculate the excess mean-square error caused by the noisy estimate of the gradient employed by the LMS algorithm. As can be noted, $\text{cov}[\Delta\mathbf{w}(k+1)]$ does not tend to $\mathbf{0}$ as $k \to \infty$, due to the last term in (3.23) that provides an excitation in the dynamic matrix equation.

A more useful form for (3.23) can be obtained by premultiplying and postmultiplying it by $\mathbf{Q}^T$ and $\mathbf{Q}$, respectively, yielding

$$\begin{aligned}\mathbf{Q}^T\text{cov}[\Delta\mathbf{w}(k+1)]\mathbf{Q} = {}& \mathbf{Q}^T\,\text{cov}[\Delta\mathbf{w}(k)]\,\mathbf{Q}\\ &- 2\mu\mathbf{Q}^T\mathbf{R}\mathbf{Q}\mathbf{Q}^T\,\text{cov}[\Delta\mathbf{w}(k)]\mathbf{Q}\\ &- 2\mu\mathbf{Q}^T\,\text{cov}[\Delta\mathbf{w}(k)]\mathbf{Q}\mathbf{Q}^T\mathbf{R}\mathbf{Q}\\ &+ 8\mu^2\mathbf{Q}^T\mathbf{R}\mathbf{Q}\mathbf{Q}^T\,\text{cov}[\Delta\mathbf{w}(k)]\mathbf{Q}\mathbf{Q}^T\mathbf{R}\mathbf{Q}\\ &+ 4\mu^2\mathbf{Q}^T\mathbf{R}\mathbf{Q}\mathbf{Q}^T\,\text{tr}\{\mathbf{R}\mathbf{Q}\mathbf{Q}^T\,\text{cov}[\Delta\mathbf{w}(k)]\}\mathbf{Q}\\ &+ 4\mu^2\sigma_n^2\mathbf{Q}^T\mathbf{R}\mathbf{Q}\end{aligned} \tag{3.25}$$

where we used the equality $\mathbf{Q}^T\mathbf{Q} = \mathbf{Q}\mathbf{Q}^T = \mathbf{I}$. Using the fact that $\mathbf{Q}^T\text{tr}[\mathbf{B}]\mathbf{Q} = \text{tr}[\mathbf{Q}^T\mathbf{B}\mathbf{Q}]\mathbf{I}$ for any $\mathbf{B}$,

$$\begin{aligned}\text{cov}[\Delta\mathbf{w}'(k+1)] = {}& \text{cov}[\Delta\mathbf{w}'(k)] - 2\mu\boldsymbol{\Lambda}\,\text{cov}[\Delta\mathbf{w}'(k)] - 2\mu\,\text{cov}[\Delta\mathbf{w}'(k)]\boldsymbol{\Lambda}\\ &+ 8\mu^2\boldsymbol{\Lambda}\,\text{cov}[\Delta\mathbf{w}'(k)]\boldsymbol{\Lambda} + 4\mu^2\boldsymbol{\Lambda}\,\text{tr}\left\{\boldsymbol{\Lambda}\,\text{cov}[\Delta\mathbf{w}'(k)]\right\} + 4\mu^2\sigma_n^2\boldsymbol{\Lambda}\end{aligned} \tag{3.26}$$

where $\text{cov}[\Delta\mathbf{w}'(k)] = E[\mathbf{Q}^T\Delta\mathbf{w}(k)\Delta\mathbf{w}^T(k)\mathbf{Q}]$.

As will be shown in Sect. 3.3.6, only the diagonal elements of $\text{cov}[\Delta\mathbf{w}'(k)]$ contribute to the excess MSE in the LMS algorithm. By defining $\mathbf{v}'(k)$ as a vector with elements consisting of the diagonal elements of $\text{cov}[\Delta\mathbf{w}'(k)]$, and $\boldsymbol{\lambda}$ as a vector consisting of the eigenvalues of $\mathbf{R}$, the following relation can be derived from the above equations

$$\begin{aligned}\mathbf{v}'(k+1) &= (\mathbf{I} - 4\mu\boldsymbol{\Lambda} + 8\mu^2\boldsymbol{\Lambda}^2 + 4\mu^2\boldsymbol{\lambda}\boldsymbol{\lambda}^T)\mathbf{v}'(k) + 4\mu^2\sigma_n^2\boldsymbol{\lambda}\\ &= \mathbf{B}\mathbf{v}'(k) + 4\mu^2\sigma_n^2\boldsymbol{\lambda}\end{aligned} \tag{3.27}$$

where the elements of **B** are given by

$$b_{ij} = \begin{cases} 1 - 4\mu\lambda_i + 8\mu^2\lambda_i^2 + 4\mu^2\lambda_i^2 & \text{for } i = j \\ 4\mu^2\lambda_i\lambda_j & \text{for } i \neq j\,. \end{cases} \tag{3.28}$$

The value of the convergence factor μ must be chosen in a range that guarantees the convergence of $\mathbf{v}'(k)$. Since matrix **B** is symmetric, it has only real-valued eigenvalues. Also since all entries of **B** are also non-negative, the maximum among the sum of elements in any row of **B** represents an upper bound to the maximum eigenvalue of **B** and to the absolute value of any other eigenvalue, see pages 53 and 63 of [14] or the Gershgorin theorem in [15]. As a consequence, a sufficient condition to guarantee convergence is to force the sum of the elements in any row of **B** to be kept in the range $0 < \sum_{j=0}^{N} b_{ij} < 1$. Since

$$\sum_{j=0}^{N} b_{ij} = 1 - 4\mu\lambda_i + 8\mu^2\lambda_i^2 + 4\mu^2\lambda_i \sum_{j=0}^{N} \lambda_j \tag{3.29}$$

the critical values of μ are those for which the above equation approaches 1, as for any μ the expression is always positive. This will occur only if the last three terms of (3.29) approach zero, that is

$$-4\mu\lambda_i + 8\mu^2\lambda_i^2 + 4\mu^2\lambda_i \sum_{j=0}^{N} \lambda_j \approx 0$$

After simple manipulation the stability condition obtained is

$$0 < \mu < \frac{1}{2\lambda_{\max} + \sum_{j=0}^{N} \lambda_j} < \frac{1}{\sum_{j=0}^{N} \lambda_j} = \frac{1}{\text{tr}[\mathbf{R}]} \tag{3.30}$$

where the last and simpler expression is more widely used in practice because tr[**R**] is quite simple to estimate since it is related to the Euclidean norm squared of the input signal vector, whereas an estimate $\lambda_{\max}$ is much more difficult to obtain. It will be shown in (3.45) that μ controls the speed of convergence of the MSE.

The upper bound obtained for the value of μ is important from the practical point of view, because it gives us an indication of the maximum value of μ that could be used in order to achieve convergence of the coefficients. However, the reader should be advised that the given upper bound is somewhat optimistic due to the approximations and assumptions made. In most cases, the value of μ should not be chosen close to the upper bound.

3.3.4 Behavior of the Error Signal

In this subsection, the mean value of the output error in the adaptive filter is calculated, considering that the unknown system model has infinite impulse response and there is measurement noise. The error signal, when an additional measurement noise is accounted for, is given by

$$e(k) = d'(k) - \mathbf{w}^T(k)\mathbf{x}(k) + n(k) \tag{3.31}$$

where $d'(k)$ is the desired signal without measurement noise. For a given known input vector $\mathbf{x}(k)$, the expected value of the error signal is

$$\begin{aligned} E[e(k)] &= E[d'(k)] - E[\mathbf{w}^T(k)\mathbf{x}(k)] + E[n(k)] \\ &= E[d'(k)] - \mathbf{w}_o^T\mathbf{x}(k) + E[n(k)] \end{aligned} \tag{3.32}$$

where $\mathbf{w}_o$ is the optimal solution, i.e., the Wiener solution for the coefficient vector. Note that the input signal vector was assumed known in the above equation, in order to expose what can be expected if the adaptive filter converges to the optimal solution. If $d'(k)$ was generated through an infinite impulse response system, a residue error remains in the subtraction of the first two terms due to undermodeling (adaptive FIR filter with insufficient number of coefficients), i.e.,

$$E[e(k)] = E\left[\sum_{i=N+1}^{\infty} h(i)x(k-i)\right] + E[n(k)] \tag{3.33}$$

where $h(i)$, for $i = N+1, \ldots, \infty$, are the coefficients of the process that generated the part of $d'(k)$ not identified by the adaptive filter. If the input signal and $n(k)$ have zero mean, then $E[e(k)] = 0$.

3.3.5 Minimum Mean-Square Error

In this subsection, the minimum MSE is calculated for undermodeling situations and in the presence of additional noise. Let's assume again the undermodeling case where the adaptive filter has less coefficients than the unknown system in a system identification setup. In this case we can write

$$\begin{aligned} d(k) &= \mathbf{h}^T\mathbf{x}_\infty(k) + n(k) \\ &= \left[\mathbf{w}_o^T \ \overline{\mathbf{h}}^T\right]\left[\begin{array}{c}\mathbf{x}(k) \\ \overline{\mathbf{x}}_\infty(k)\end{array}\right] + n(k) \end{aligned} \tag{3.34}$$

where $\mathbf{w}_o$ is a vector containing the first $N+1$ coefficients of the unknown system impulse response, $\overline{\mathbf{h}}$ contains the remaining elements of $\mathbf{h}$. The output signal of an adaptive filter with $N+1$ coefficients is given by

$$y(k) = \mathbf{w}^T(k)\mathbf{x}(k)$$

In this setup the MSE has the following expression

$$\begin{aligned}
\xi &= E\Big\{d^2(k) - 2\mathbf{w}_o^T\mathbf{x}(k)\mathbf{w}^T(k)\mathbf{x}(k) - 2\overline{\mathbf{h}}^T\overline{\mathbf{x}}_\infty(k)\mathbf{w}^T(k)\mathbf{x}(k) \\
&\qquad -2[\mathbf{w}^T(k)\mathbf{x}(k)]n(k) + [\mathbf{w}^T(k)\mathbf{x}(k)]^2\Big\} \\
&= E\left\{d^2(k) - 2[\mathbf{w}^T(k)\ \mathbf{0}_\infty^T]\begin{bmatrix}\mathbf{x}(k)\\ \overline{\mathbf{x}}_\infty(k)\end{bmatrix}[\mathbf{w}_o^T\ \overline{\mathbf{h}}^T]\begin{bmatrix}\mathbf{x}(k)\\ \overline{\mathbf{x}}_\infty(k)\end{bmatrix}\right. \\
&\qquad \left. -2[\mathbf{w}^T(k)\mathbf{x}(k)]n(k) + [\mathbf{w}^T(k)\mathbf{x}(k)]^2\right\} \\
&= E[d^2(k)] - 2[\mathbf{w}^T(k)\ \mathbf{0}_\infty^T]\mathbf{R}_\infty\begin{bmatrix}\mathbf{w}_o\\ \overline{\mathbf{h}}\end{bmatrix} + \mathbf{w}^T(k)\mathbf{R}\mathbf{w}(k) \qquad (3.35)
\end{aligned}$$

where

$$\mathbf{R}_\infty = E\left\{\begin{bmatrix}\mathbf{x}(k)\\ \overline{\mathbf{x}}_\infty(k)\end{bmatrix}[\mathbf{x}^T(k)\ \overline{\mathbf{x}}_\infty^T(k)]\right\}$$

and $\mathbf{0}_\infty$ is an infinite length vector whose elements are zeros. By calculating the derivative of ξ with respect to the coefficients of the adaptive filter, it follows that (see derivations around (2.91) and (2.148))

$$\begin{aligned}
\hat{\mathbf{w}}_o &= \mathbf{R}^{-1}\text{trunc}\,\{\mathbf{p}_\infty\}_{N+1} = \mathbf{R}^{-1}\text{trunc}\left\{\mathbf{R}_\infty\begin{bmatrix}\mathbf{w}_o\\ \overline{\mathbf{h}}\end{bmatrix}\right\}_{N+1} \\
&= \mathbf{R}^{-1}\text{trunc}\{\mathbf{R}_\infty\mathbf{h}\}_{N+1} \qquad (3.36)
\end{aligned}$$

where $\text{trunc}\{\mathbf{a}\}_{N+1}$ represents a vector generated by retaining the first $N+1$ elements of $\mathbf{a}$. It should be noticed that the results of (3.35) and (3.36) are algorithm independent.

The minimum mean-square error can be obtained from (3.35), when assuming the input signal is a white noise uncorrelated with the additional noise signal, that is

$$\begin{aligned}
\xi_{\min} = E[e^2(k)]_{\min} &= \sum_{i=N+1}^{\infty} h^2(i)E[x^2(k-i)] + E[n^2(k)] \\
&= \sum_{i=N+1}^{\infty} h^2(i)\sigma_x^2 + \sigma_n^2 \qquad (3.37)
\end{aligned}$$

This minimum error is achieved when it is assumed that the adaptive-filter multiplier coefficients are frozen at their optimum values, refer to (2.148) for similar discussion. In case the adaptive filter has sufficient order to model the process that generated $d(k)$, the minimum MSE that can be achieved is equal to the variance of the additional noise, given by σ_n^2. The reader should note that the effect of undermodeling discussed in this subsection generates an excess MSE with respect to σ_n^2.

3.3.6 Excess Mean-Square Error and Misadjustment

The result of the previous subsection assumes that the adaptive-filter coefficients converge to their optimal values, but in practice this is not so. Although the coefficient vector on average converges to $\mathbf{w}_o$, the instantaneous deviation $\Delta\mathbf{w}(k) = \mathbf{w}(k) - \mathbf{w}_o$, caused by the noisy gradient estimates, generates an excess MSE. The excess MSE can be quantified as described in the present subsection. The output error at instant k is given by

$$\begin{aligned} e(k) &= d(k) - \mathbf{w}_o^T\mathbf{x}(k) - \Delta\mathbf{w}^T(k)\mathbf{x}(k) \\ &= e_o(k) - \Delta\mathbf{w}^T(k)\mathbf{x}(k) \end{aligned} \tag{3.38}$$

then

$$e^2(k) = e_o^2(k) - 2e_o(k)\Delta\mathbf{w}^T(k)\mathbf{x}(k) + \Delta\mathbf{w}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)\Delta\mathbf{w}(k) \tag{3.39}$$

The so-called independence theory assumes that the vectors $\mathbf{x}(k)$, for all k, are statistically independent, allowing a simple mathematical treatment for the LMS algorithm. As mentioned before, this assumption is in general not true, especially in the case where $\mathbf{x}(k)$ consists of the elements of a delay line. However, even in this case the use of the independence assumption is justified by the agreement between the analytical and the experimental results. With the independence assumption, $\Delta\mathbf{w}(k)$ can be considered independent of $\mathbf{x}(k)$, since only previous input vectors are involved in determining $\Delta\mathbf{w}(k)$. By using the assumption and applying the expected value operator to (3.39), we have

$$\begin{aligned} \xi(k) &= E[e^2(k)] \\ &= \xi_{\min} - 2E[\Delta\mathbf{w}^T(k)]E[e_o(k)\mathbf{x}(k)] + E[\Delta\mathbf{w}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)\Delta\mathbf{w}(k)] \\ &= \xi_{\min} - 2E[\Delta\mathbf{w}^T(k)]E[e_o(k)\mathbf{x}(k)] + E\left\{\mathrm{tr}[\Delta\mathbf{w}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)\Delta\mathbf{w}(k)]\right\} \\ &= \xi_{\min} - 2E[\Delta\mathbf{w}^T(k)]E[e_o(k)\mathbf{x}(k)] + E\left\{\mathrm{tr}[\mathbf{x}(k)\mathbf{x}^T(k)\Delta\mathbf{w}(k)\Delta\mathbf{w}^T(k)]\right\} \end{aligned} \tag{3.40}$$

where in the fourth equality we used the property $\mathrm{tr}[\mathbf{A}\cdot\mathbf{B}] = \mathrm{tr}[\mathbf{B}\cdot\mathbf{A}]$. The last term of the above equation can be rewritten as

$$\mathrm{tr}\left\{E[\mathbf{x}(k)\mathbf{x}^T(k)]E[\Delta\mathbf{w}(k)\Delta\mathbf{w}^T(k)]\right\}$$

Since $\mathbf{R} = E[\mathbf{x}(k)\mathbf{x}^T(k)]$ and by the orthogonality principle $E[e_o(k)\mathbf{x}(k)] = 0$, the above equation can be simplified as follows:

$$\xi(k) = \xi_{\min} + E[\Delta\mathbf{w}^T(k)\mathbf{R}\Delta\mathbf{w}(k)] \tag{3.41}$$

The excess in the MSE is given by

$$\begin{aligned}\Delta\xi(k) \overset{\triangle}{=} \xi(k) - \xi_{\min} &= E[\Delta\mathbf{w}^T(k)\mathbf{R}\Delta\mathbf{w}(k)] \\ &= E\{\text{tr}[\mathbf{R}\Delta\mathbf{w}(k)\Delta\mathbf{w}^T(k)]\} \\ &= \text{tr}\{E[\mathbf{R}\Delta\mathbf{w}(k)\Delta\mathbf{w}^T(k)]\}\end{aligned} \tag{3.42}$$

By using the fact that $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}$, the following relation results

$$\begin{aligned}\Delta\xi(k) &= \text{tr}\left\{E[\mathbf{Q}\mathbf{Q}^T\mathbf{R}\mathbf{Q}\mathbf{Q}^T\Delta\mathbf{w}(k)\Delta\mathbf{w}^T(k)\mathbf{Q}\mathbf{Q}^T]\right\} \\ &= \text{tr}\{\mathbf{Q}\boldsymbol{\Lambda}\ \text{cov}[\Delta\mathbf{w}'(k)]\mathbf{Q}^T\}\end{aligned} \tag{3.43}$$

Therefore,

$$\Delta\xi(k) = \text{tr}\{\boldsymbol{\Lambda}\ \text{cov}[\Delta\mathbf{w}'(k)]\} \tag{3.44}$$

From (3.27), it is possible to show that

$$\Delta\xi(k) = \sum_{i=0}^{N} \lambda_i v_i'(k) = \boldsymbol{\lambda}^T\mathbf{v}'(k) \tag{3.45}$$

Since

$$v_i'(k+1) = (1 - 4\mu\lambda_i + 8\mu^2\lambda_i^2)v_i'(k) + 4\mu^2\lambda_i \sum_{j=0}^{N} \lambda_j v_j'(k) + 4\mu^2\sigma_n^2\lambda_i \tag{3.46}$$

and $v_i'(k+1) \approx v_i'(k)$ for large k, we can apply a summation operation to the above equation in order to obtain

$$\begin{aligned}\sum_{j=0}^{N} \lambda_j v_j'(k) &= \frac{\mu\sigma_n^2 \sum_{i=0}^{N} \lambda_i + 2\mu \sum_{i=0}^{N} \lambda_i^2 v_i'(k)}{1 - \mu \sum_{i=0}^{N} \lambda_i} \\ &\approx \frac{\mu\sigma_n^2 \sum_{i=0}^{N} \lambda_i}{1 - \mu \sum_{i=0}^{N} \lambda_i} \\ &= \frac{\mu\sigma_n^2 \text{tr}[\mathbf{R}]}{1 - \mu \text{tr}[\mathbf{R}]}\end{aligned} \tag{3.47}$$

where the term $2\mu \sum_{i=0}^{N} \lambda_i^2 v_i'(k)$ was considered very small as compared to the remaining terms of the numerator. This assumption is not easily justifiable, but is valid for small values of μ.

The excess mean-square error can then be expressed as

$$\xi_{\text{exc}} = \lim_{k\to\infty} \Delta\xi(k) \approx \frac{\mu\sigma_n^2 \text{tr}[\mathbf{R}]}{1-\mu\text{tr}[\mathbf{R}]} \tag{3.48}$$

This equation, for very small μ, can be approximated by

$$\xi_{\text{exc}} \approx \mu\sigma_n^2 \text{tr}[\mathbf{R}] = \mu(N+1)\sigma_n^2\sigma_x^2 \tag{3.49}$$

where σ_x^2 is the input signal variance and σ_n^2 is the additional-noise variance.

The misadjustment M, defined as the ratio between the ξ_{exc} and the minimum MSE, is a common parameter used to compare different adaptive signal processing algorithms. For the LMS algorithm, the misadjustment is given by

$$M \triangleq \frac{\xi_{\text{exc}}}{\xi_{\text{min}}} \approx \frac{\mu\text{tr}[\mathbf{R}]}{1-\mu\text{tr}[\mathbf{R}]} \tag{3.50}$$

3.3.7 *Transient Behavior*

Before the LMS algorithm reaches the steady-state behavior, a number of iterations are spent in the transient part. During this time, the adaptive-filter coefficients and the output error change from their initial values to values close to that of the corresponding optimal solution.

In the case of the adaptive-filter coefficients, the convergence in the mean will follow $(N+1)$ geometric decaying curves with ratios $r_{wi} = (1-2\mu\lambda_i)$. Each of these curves can be approximated by an exponential envelope with time constant τ_{wi} as follows (see (3.18)) [2]:

$$r_{wi} = \mathrm{e}^{\frac{-1}{\tau_{wi}}} = 1 - \frac{1}{\tau_{wi}} + \frac{1}{2!\tau_{wi}^2} + \cdots \tag{3.51}$$

where for each iteration, the decay in the exponential envelope is equal to the decay in the original geometric curve. In general, r_{wi} is slightly smaller than one, especially for the slowly decreasing modes corresponding to small λ_i and μ. Therefore,

$$r_{wi} = (1-2\mu\lambda_i) \approx 1 - \frac{1}{\tau_{wi}} \tag{3.52}$$

then

$$\tau_{wi} = \frac{1}{2\mu\lambda_i}$$

for $i = 0, 1, \ldots, N$. Note that in order to guarantee convergence of the tap coefficients in the mean, μ must be chosen in the range $0 < \mu < 1/\lambda_{\max}$ (see (3.19)).

According to (3.30), for the convergence of the MSE the range of values for μ is $0 < \mu < 1/\text{tr}[\mathbf{R}]$, and the corresponding time constant can be calculated from matrix $\mathbf{B}$ in (3.27), by considering the terms in μ^2 small as compared to the remaining terms in matrix $\mathbf{B}$. In this case, the geometric decaying curves have ratios given by $r_{ei} = (1 - 4\mu\lambda_i)$ that can be fitted to exponential envelopes with time constants given by

$$\tau_{ei} = \frac{1}{4\mu\lambda_i} \tag{3.53}$$

for $i = 0, 1, \ldots, N$. In the convergence of both the error and the coefficients, the time required for the convergence depends on the ratio of eigenvalues of the input signal correlation matrix.

Returning to the tap coefficients case, if μ is chosen to be approximately $1/\lambda_{\max}$ the corresponding time constant for the coefficients is given by

$$\tau_{wi} \approx \frac{\lambda_{\max}}{2\lambda_i} \leq \frac{\lambda_{\max}}{2\lambda_{\min}} \tag{3.54}$$

Since the mode with the highest time constant takes longer to reach convergence, the rate of convergence is determined by the slowest mode given by $\tau_{w_{\max}} = \lambda_{\max}/(2\lambda_{\min})$. Suppose the convergence is considered achieved when the slowest mode provides an attenuation of 100, i.e.,

$$\mathrm{e}^{\frac{-k}{\tau_{w\max}}} = 0.01$$

this requires the following number of iterations in order to reach convergence:

$$k \approx 4.6\frac{\lambda_{\max}}{2\lambda_{\min}}$$

The above situation is quite optimistic because μ was chosen to be high. As mentioned before, in practice we should choose the value of μ much smaller than the upper bound. For an eigenvalue spread approximating one, according to (3.30) let's choose μ smaller than $1/[(N + 3)\lambda_{\max}]$.[2] In this case, the LMS algorithm will require at least

$$k \approx 4.6\frac{(N+3)\lambda_{\max}}{2\lambda_{\min}} \approx 2.3(N+3)$$

iterations to achieve convergence in the coefficients.

[2] This choice also guarantees the convergence of the MSE.

The analytical results presented in this section are valid for stationary environments. The LMS algorithm can also operate in the case of nonstationary environments, as shown in the following section.

3.4 LMS Algorithm Behavior in Nonstationary Environments

In practical situations, the environment in which the adaptive filter is embedded may be nonstationary. In these cases, the input signal autocorrelation matrix and/or the cross-correlation vector, denoted, respectively, by $\mathbf{R}(k)$ and $\mathbf{p}(k)$, are/is varying with time. Therefore, the optimal solution for the coefficient vector is also a time-varying vector given by $\mathbf{w}_o(k)$.

Since the optimal coefficient vector is not fixed, it is important to analyze if the LMS algorithm will be able to track changes in $\mathbf{w}_o(k)$. It is also of interest to learn how the tracking error in the coefficients given by $E[\mathbf{w}(k)] - \mathbf{w}_o(k)$ will affect the output MSE. It will be shown later that the excess MSE caused by lag in the tracking of $\mathbf{w}_o(k)$ can be separated from the excess MSE caused by the measurement noise, and therefore, without loss of generality, in the following analysis the additional noise will be considered zero.

The coefficient-vector updating in the LMS algorithm can be written in the following form

$$\begin{aligned}\mathbf{w}(k+1) &= \mathbf{w}(k) + 2\mu\mathbf{x}(k)e(k)\\ &= \mathbf{w}(k) + 2\mu\mathbf{x}(k)[d(k) - \mathbf{x}^T(k)\mathbf{w}(k)]\end{aligned} \tag{3.55}$$

Since

$$d(k) = \mathbf{x}^T(k)\mathbf{w}_o(k) \tag{3.56}$$

the coefficient updating can be expressed as follows:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + 2\mu\mathbf{x}(k)[\mathbf{x}^T(k)\mathbf{w}_o(k) - \mathbf{x}^T(k)\mathbf{w}(k)] \tag{3.57}$$

Now assume that an ensemble of a nonstationary adaptive identification process has been built, where the input signal in each experiment is taken from the same stochastic process. The input signal is considered stationary. This assumption results in a fixed $\mathbf{R}$ matrix, and the nonstationarity is caused by the desired signal that is generated by applying the input signal to a time-varying system. With these assumptions, by using the expected value operation to the ensemble, with the coefficient updating in each experiment given by (3.57), and additionally assuming that $\mathbf{w}(k)$ is independent of $\mathbf{x}(k)$ yields

$$\begin{aligned}E[\mathbf{w}(k+1)] &= E[\mathbf{w}(k)] + 2\mu E[\mathbf{x}(k)\mathbf{x}^T(k)]\mathbf{w}_o(k) - 2\mu E[\mathbf{x}(k)\mathbf{x}^T(k)]E[\mathbf{w}(k)]\\ &= E[\mathbf{w}(k)] + 2\mu\mathbf{R}\{\mathbf{w}_o(k) - E[\mathbf{w}(k)]\}\end{aligned} \tag{3.58}$$

If the lag in the coefficient vector is defined by

$$\mathbf{l_W}(k) = E[\mathbf{w}(k)] - \mathbf{w}_o(k) \tag{3.59}$$

(3.58) can be rewritten as

$$\mathbf{l_W}(k+1) = (\mathbf{I} - 2\mu\mathbf{R})\mathbf{l_W}(k) - \mathbf{w}_o(k+1) + \mathbf{w}_o(k) \tag{3.60}$$

In order to simplify our analysis, we can premultiply the above equation by $\mathbf{Q}^T$, resulting in a decoupled set of equations given by

$$\mathbf{l}'_\mathbf{W}(k+1) = (\mathbf{I} - 2\mu\boldsymbol{\Lambda})\mathbf{l}'_\mathbf{W}(k) - \mathbf{w}'_o(k+1) + \mathbf{w}'_o(k) \tag{3.61}$$

where the vectors with superscript are the original vectors projected onto the transformed space. As can be noted, each element of the lag-error vector is determined by the following relation

$$l'_i(k+1) = (1 - 2\mu\lambda_i)l'_i(k) - w'_{oi}(k+1) + w'_{oi}(k) \tag{3.62}$$

where $l'_i(k)$ is the ith element of $\mathbf{l}'_\mathbf{W}(k)$. By properly interpreting the above equation, we can say that the lag is generated by applying the transformed instantaneous optimal coefficient to a first-order discrete-time *lag filter* denoted as $L''_i(z)$, i.e.,

$$L'_i(z) = -\frac{z-1}{z-1+2\mu\lambda_i}W'_{oi}(z) = L''_i(z)W'_{oi}(z) \tag{3.63}$$

The discrete-time filter transient response converges with a time constant of the exponential envelope given by

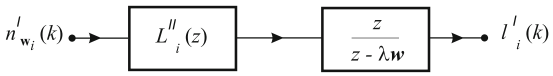
$$\tau_i = \frac{1}{2\mu\lambda_i} \tag{3.64}$$

which is of course different for each individual tap. Therefore, the tracking ability of the coefficients in the LMS algorithm is dependent on the eigenvalues of the input signal correlation matrix.

The lag in the adaptive-filter coefficients leads to an excess mean-square error. In order to calculate the excess MSE, suppose that each element of the optimal coefficient vector is modeled as a first-order Markov process. This nonstationary situation can be considered somewhat simplified as compared with some real practical situations. However, it allows a manageable mathematical analysis while retaining the essence of handling the more complicated cases. The first-order Markov process is described by

$$\mathbf{w}_o(k) = \lambda_\mathbf{W}\mathbf{w}_o(k-1) + \mathbf{n_W}(k) \tag{3.65}$$

Fig. 3.2 Lag model in nonstationary environment



where $\mathbf{n_w}(k)$ is a vector whose elements are zero-mean white noise processes with variance $\sigma_{\mathbf{w}}^2$, and $\lambda_{\mathbf{w}} < 1$. Note that $(1 - 2\mu\lambda_i) < \lambda_{\mathbf{w}} < 1$, for $i = 0, 1, \ldots, N$, since the optimal coefficients values must vary slower than the adaptive-filter tracking speed, i.e., $\frac{1}{2\mu\lambda_i} < \frac{1}{1-\lambda_{\mathbf{w}}}$. This model may not represent an actual system when $\lambda_{\mathbf{w}} \rightarrow 1$, since the $E[\mathbf{w}_o(k)\mathbf{w}_o^T(k)]$ will have unbounded elements if, for example, $\mathbf{n_w}(k)$ is not exactly zero mean. A more realistic model would include a factor $(1 - \lambda_{\mathbf{w}})^{\frac{p}{2}}$, for $p \geq 1$, multiplying $\mathbf{n_w}(k)$ in order to guarantee that $E[\mathbf{w}_o(k)\mathbf{w}_o^T(k)]$ is bounded. In the following discussions, this case will not be considered since the corresponding results can be easily derived (see Problem 14).

From (3.62) and (3.63), we can infer that the lag-error vector elements are generated by applying a first-order discrete-time system to the elements of the unknown system coefficient vector, both in the transformed space. On the other hand, the coefficients of the unknown system are generated by applying each element of the noise vector $\mathbf{n_w}(k)$ to a first-order all-pole filter, with the pole placed at $\lambda_{\mathbf{w}}$. For the unknown coefficient vector with the above model, the lag-error vector elements can be generated by applying each element of the transformed noise vector $\mathbf{n}'_{\mathbf{w}}(k) = \mathbf{Q}^T \mathbf{n_w}(k)$ to a discrete-time filter with transfer function

$$H_i(z) = \frac{-(z-1)z}{(z - 1 + 2\mu\lambda_i)(z - \lambda_{\mathbf{w}})} \tag{3.66}$$

This transfer function consists of a cascade of the lag filter $L_i''(z)$ with the all-pole filter representing the first-order Markov process as illustrated in Fig. 3.2. Using the inverse $\mathcal{Z}$-transform, the variance of the elements of the vector $\mathbf{l}'_{\mathbf{w}}(k)$ can then be calculated by

$$\begin{aligned} E[l_i'^2(k)] &= \frac{1}{2\pi J} \oint H_i(z) H_i(z^{-1}) \sigma_{\mathbf{w}}^2 z^{-1} \, dz \\ &= \left[\frac{1}{(1 - \lambda_{\mathbf{w}} - 2\mu\lambda_i)(1 - \lambda_{\mathbf{w}} + 2\mu\lambda_i\lambda_{\mathbf{w}})} \right] \left[\frac{-\mu\lambda_i}{1 - \mu\lambda_i} + \frac{1 - \lambda_{\mathbf{w}}}{1 + \lambda_{\mathbf{w}}} \right] \sigma_{\mathbf{w}}^2 \end{aligned} \tag{3.67}$$

If $\lambda_{\mathbf{w}}$ is considered very close to 1, it is possible to simplify the above equation as

$$E[l_i'^2(k)] \approx \frac{\sigma_{\mathbf{w}}^2}{4\mu\lambda_i(1 - \mu\lambda_i)} \tag{3.68}$$

Any error in the coefficient vector of the adaptive filter as compared to the optimal coefficient filter generates an excess MSE (see (3.41)). Since the lag is one source of error in the adaptive-filter coefficients, then the excess MSE due to lag is given by

$$\begin{aligned}\xi_{\text{lag}} &= E[\mathbf{l}_{\mathbf{w}}^T(k)\mathbf{R}\mathbf{l}_{\mathbf{w}}(k)] \\ &= E\left\{\text{tr}[\mathbf{R}\mathbf{l}_{\mathbf{w}}(k)\mathbf{l}_{\mathbf{w}}^T(k)]\right\} \\ &= \text{tr}\left\{\mathbf{R}E[\mathbf{l}_{\mathbf{w}}(k)\mathbf{l}_{\mathbf{w}}^T(k)]\right\} \\ &= \text{tr}\left\{\boldsymbol{\Lambda}E[\mathbf{l}'_{\mathbf{w}}(k)\mathbf{l}'^T_{\mathbf{w}}(k)]\right\} \\ &= \sum_{i=0}^{N}\lambda_i E[l_i'^2(k)] \\ &\approx \frac{\sigma_{\mathbf{w}}^2}{4\mu}\sum_{i=0}^{N}\frac{1}{1-\mu\lambda_i}\end{aligned} \tag{3.69}$$

If μ is very small, the MSE due to lag tends to infinity indicating that the LMS algorithm, in this case, cannot track any change in the environment. On the other hand, for μ appropriately chosen the algorithm can track variations in the environment leading to an excess MSE. This excess MSE depends on the variance of the optimal coefficient disturbance and on the values of the input signal autocorrelation matrix eigenvalues, as indicated in (3.69). In the case μ is very small and $\lambda_{\mathbf{w}}$ is not very close to 1, the approximation for (3.67) becomes

$$E[l_i'^2(k)] \approx \frac{\sigma_{\mathbf{w}}^2}{1-\lambda_{\mathbf{w}}^2} \tag{3.70}$$

As a result the MSE due to lag is given by

$$\xi_{\text{lag}} \approx \frac{(N+1)\sigma_{\mathbf{w}}^2}{1-\lambda_{\mathbf{w}}^2} \tag{3.71}$$

It should be noticed that $\lambda_{\mathbf{w}}$ closer to 1 than the modes of the adaptive filter is the common operation region, therefore the result of (3.71) is not discussed further.

Now we analyze how the error due to lag interacts with the error generated by the noisy calculation of the gradient in the LMS algorithm. The overall error in the taps is given by

$$\Delta\mathbf{w}(k) = \mathbf{w}(k) - \mathbf{w}_o(k) = \{\mathbf{w}(k) - E[\mathbf{w}(k)]\} + \{E[\mathbf{w}(k)] - \mathbf{w}_o(k)\} \tag{3.72}$$

where the first error in the above equation is due to the additional noise and the second is the error due to lag. The overall excess MSE can then be expressed as

$$\begin{aligned}\xi_{\text{total}} &= E\{[\mathbf{w}(k)-\mathbf{w}_o(k)]^T\mathbf{R}[\mathbf{w}(k)-\mathbf{w}_o(k)]\} \\ &\approx E\{(\mathbf{w}(k)-E[\mathbf{w}(k)])^T\mathbf{R}(\mathbf{w}(k)-E[\mathbf{w}(k)])\} \\ &\quad + E\{(E[\mathbf{w}(k)]-\mathbf{w}_o(k))^T\mathbf{R}(E[\mathbf{w}(k)]-\mathbf{w}_o(k))\}\end{aligned} \tag{3.73}$$

since $2E\{(\mathbf{w}(k) - E[\mathbf{w}(k)])^T \mathbf{R}(E[\mathbf{w}(k)] - \mathbf{w}_o(k))\} \approx 0$, if we consider the fact that $\mathbf{w}_o(k)$ is kept fixed in each experiment of the ensemble. As a consequence, an estimate for the overall excess MSE can be obtained by adding the results of (3.48) and (3.69), i.e.,

$$\xi_{\text{total}} \approx \frac{\mu\sigma_n^2 \text{tr}[\mathbf{R}]}{1 - \mu\text{tr}[\mathbf{R}]} + \frac{\sigma_{\mathbf{w}}^2}{4\mu} \sum_{i=0}^{N} \frac{1}{1 - \mu\lambda_i} \tag{3.74}$$

If small μ is employed, the above equation can be simplified as follows:

$$\xi_{\text{total}} \approx \mu\sigma_n^2 \text{tr}[\mathbf{R}] + \frac{\sigma_{\mathbf{w}}^2}{4\mu}(N+1) \tag{3.75}$$

Differentiating the above equation with respect to μ and setting the result to zero yields an optimum value for μ given by

$$\mu_{\text{opt}} = \sqrt{\frac{(N+1)\sigma_{\mathbf{w}}^2}{4\sigma_n^2 \text{tr}[\mathbf{R}]}} \tag{3.76}$$

The μ_{opt} is supposed to lead to the minimum excess MSE. However, the user should bear in mind that the μ_{opt} can only be used if it satisfies stability conditions, and if its value can be considered small enough to validate (3.75). Also this value is optimum only when quantization effects are not taken into consideration, where for short-wordlength implementation the best μ should be chosen following the guidelines given in Chap. 15. It should also be mentioned that the study of the misadjustment due to nonstationarity of the environment is considerably more complicated when the input signal and the desired signal are simultaneously nonstationary [8, 10–17]. Therefore, the analysis presented here is only valid if the assumptions made are valid. However, the simplified analysis provides a good sample of the LMS algorithm behavior in a nonstationary environment and gives a general indication of what can be expected in more complicated situations.

The results of the analysis of the previous sections are obtained assuming that the algorithm is implemented with infinite precision.[3] However, the widespread use of adaptive-filtering algorithms in real-time requires their implementation with short wordlength, in order to meet the speed requirements. When implemented with short-wordlength precision the LMS algorithm behavior can be very different from what is expected in infinite precision. In particular, when the convergence factor μ tends to zero it is expected that the minimum mean-square error is reached in steady state; however, due to quantization effects the MSE tends to increase significantly if μ is reduced below a certain value. In fact, the algorithm can stop updating some filter coefficients if μ is not chosen appropriately. Chapter 15, Sect. 15.1, presents detailed analysis of the quantization effects in the LMS algorithm.

[3] This is an abuse of language, by infinite precision we mean very long wordlength.

3.5 Complex LMS Algorithm

The LMS algorithm for complex signals, which often appear in communications applications, is derived in Chap. 14. References [18, 19] provide details related to complex differentiation required to generate algorithms working in environments with complex signals.

By recalling that the LMS algorithm utilizes instantaneous estimates of matrix $\mathbf{R}$, denoted by $\hat{\mathbf{R}}(k)$, and of vector $\mathbf{p}$, denoted by $\hat{\mathbf{p}}(k)$, given by

$$\begin{aligned}\hat{\mathbf{R}}(k) &= \mathbf{x}(k)\mathbf{x}^H(k)\\ \hat{\mathbf{p}}(k) &= d^*(k)\mathbf{x}(k)\end{aligned} \tag{3.77}$$

The actual objective function being minimized is the instantaneous square error $|e(k)|^2$. According to the derivations in Sect. 14.3, the expression of the gradient estimate is

$$\hat{\mathbf{g}}_{\mathbf{w}^*}\{e(k)e^*(k)\} = -e^*(k)\mathbf{x}(k) \tag{3.78}$$

By utilizing the output error definition for the complex environment case and the instantaneous gradient expression, the updating equations for the complex LMS algorithm are described by

$$\begin{cases} e(k) = d(k) - \mathbf{w}^H(k)\mathbf{x}(k)\\ \mathbf{w}(k+1) = \mathbf{w}(k) + \mu_c e^*(k)\mathbf{x}(k)\end{cases} \tag{3.79}$$

If the convergence factor $\mu_c = 2\mu$, the expressions for the coefficient updating equation of the complex and real cases have the same form and the analysis results for the real case equally applies to the complex case.[4]

An iteration of the complex LMS requires $N + 2$ complex multiplications for the filter coefficient updating and $N + 1$ complex multiplications for the error generation. In a non-optimized form each complex multiplication requires four real multiplications. The detailed description of the complex LMS algorithm is shown in the table denoted as Algorithm 3.2. As for any adaptive-filtering algorithm, the initialization is not necessarily performed as described in Algorithm 3.2, where the coefficients of the adaptive filter are started with zeros.

[4] The missing factor 2 here originates from the term $\frac{1}{2}$ in definition of the gradient that we opted to use in order to be coherent with most literature, in actual implementation the factor 2 of the real case is usually incorporated to the μ.

Algorithm 3.2 Complex LMS algorithm

```
Initialization
 $\mathbf{x}(0) = \mathbf{w}(0) = [0\,0\,\ldots\,0]^T$
Do for $k \geq 0$
 $e(k) = d(k) - \mathbf{w}^H(k)\mathbf{x}(k)$
 $\mathbf{w}(k+1) = \mathbf{w}(k) + \mu_c e^*(k)\mathbf{x}(k)$
```

3.6 Examples

In this section, a number of examples are presented in order to illustrate the use of the LMS algorithm as well as to verify theoretical results presented in the previous sections.

3.6.1 Analytical Examples

Some analytical tools presented so far are employed to characterize two interesting types of adaptive-filtering problems. The problems are also solved with the LMS algorithm.

Example 3.1. A Gaussian white noise with unit variance colored by a filter with transfer function

$$H_{in}(z) = \frac{1}{z - 0.5}$$

is transmitted through a communication channel with model given by

$$H_c(z) = \frac{1}{z + 0.8}$$

and with the channel noise being Gaussian white noise with variance $\sigma_n^2 = 0.1$.

Figure 3.3 illustrates the experimental environment. Note that $x'(k)$ is generated by first applying Gaussian white noise with variance $\sigma_{in}^2 = 1$ to a filter with transfer function $H_{in}(z)$. The result is applied to a communication channel with transfer function $H_c(z)$, and then Gaussian channel noise with variance $\sigma_n^2 = 0.1$ is added. On the other hand, $d(k)$ is generated by applying the same Gaussian noise with variance $\sigma_{in}^2 = 1$ to the filter with transfer function $H_{in}(z)$, with the result delayed by L samples.

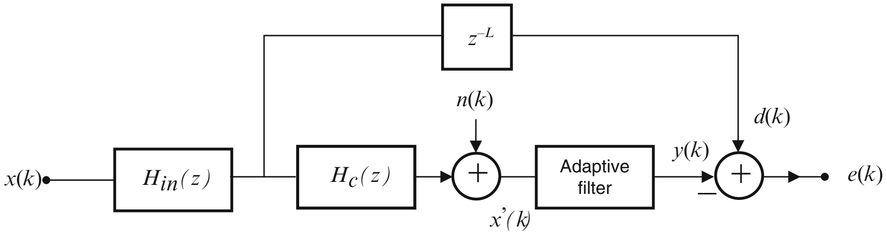


Fig. 3.3 Channel equalization of Example 3.1

(a) Determine the best value for the delay L.
(b) Compute the Wiener solution.
(c) Choose an appropriate value for μ and plot the convergence path for the LMS algorithm on the MSE surface.
(d) Plot the learning curves of the MSE and the filter coefficients in a single run as well as for the average of 25 runs.

Solution. (a) In order to determine L, we will examine the behavior of the cross-correlation between the adaptive-filter input signal denoted by $x'(k)$ and the reference signal $d(k)$.

The cross-correlation between $d(k)$ and $x'(k)$ is given by

$$\begin{aligned} p(i) &= E[d(k)x'(k-i)] \\ &= \frac{1}{2\pi \jmath} \oint H_{in}(z) z^{-L} z^{i} H_{in}(z^{-1}) H_c(z^{-1}) \sigma_{in}^2 \frac{dz}{z} \\ &= \frac{1}{2\pi \jmath} \oint \frac{1}{z-0.5} z^{-L} z^{i} \frac{z}{1-0.5z} \frac{z}{1+0.8z} \sigma_{in}^2 \frac{dz}{z} \end{aligned}$$

where the integration path is a counterclockwise closed contour corresponding to the unit circle.

The contour integral of the above equation can be solved through the Cauchy's residue theorem. For $L = 0$ and $L = 1$, the general solution is

$$p(0) = E[d(k)x'(k)] = \sigma_{in}^2 \left[0.5^{-L+1} \frac{1}{0.75} \frac{1}{1.4} \right]$$

where in order to obtain $p(0)$, we computed the residue at the pole located at 0.5. The values of the cross-correlation for $L = 0$ and $L = 1$ are, respectively

$$\begin{aligned} p(0) &= 0.47619 \\ p(0) &= 0.95238 \end{aligned}$$

For $L = 2$, we have that

$$p(0) = \sigma_{in}^2 \left[0.5^{-L+1} \frac{1}{0.75} \frac{1}{1.4} - 2 \right] = -0.09522$$

where in this case we computed the residues at the poles located at 0.5 and at 0, respectively. For $L = 3$, we have

$$p(0) = \sigma_{in}^2 [\frac{0.5^{-L+1}}{1.05} - 3.4] = 0.4095$$

From the above analysis, we see that the strongest correlation between $x'(k)$ and $d(k)$ occurs for $L = 1$. For this delay, the equalization is more effective. As a result, from the above calculations, we can obtain the elements of vector **p** as follows:

$$\mathbf{p} = \begin{bmatrix} p(0) \\ p(1) \end{bmatrix} = \begin{bmatrix} 0.9524 \\ 0.4762 \end{bmatrix}$$

Note that $p(1)$ for $L = 1$ is equal to $p(0)$ for $L = 0$.

The elements of the correlation matrix of the adaptive-filter input signal are calculated as follows:

$$\begin{aligned} r(i) &= E[x'(k)x'(k-i)] \\ &= \frac{1}{2\pi \jmath} \oint H_{in}(z) H_c(z) z^i H_{in}(z^{-1}) H_c(z^{-1}) \sigma_{in}^2 \frac{dz}{z} + \sigma_n^2 \delta(i) \\ &= \frac{1}{2\pi \jmath} \oint \frac{1}{z - 0.5} \frac{1}{z + 0.8} z^i \frac{z}{1 - 0.5z} \frac{z}{1 + 0.8z} \sigma_{in}^2 \frac{dz}{z} + \sigma_n^2 \delta(i) \end{aligned}$$

where again the integration path is a counterclockwise closed contour corresponding to the unit circle, and $\delta(i)$ is the unitary impulse. Solving the contour integral equation, we obtain

$$\begin{aligned} r(0) &= E[x'^2(k)] \\ &= \sigma_{in}^2 \left[\frac{1}{1.3} \frac{0.5}{0.75} \frac{1}{1.4} + \frac{-1}{1.3} \frac{-0.8}{1.4} \frac{1}{0.36} \right] + \sigma_n^2 = 1.6873 \end{aligned}$$

where in order to obtain $r(0)$, we computed the residues at the poles located at 0.5 and −0.8, respectively. Similarly, we have that

$$\begin{aligned} r(1) &= E[x'(k)x'(k-1)] \\ &= \sigma_{in}^2 \left[\frac{1}{1.3} \frac{1}{0.75} \frac{1}{1.4} + \frac{-1}{1.3} \frac{1}{1.4} \frac{1}{0.36} \right] = -0.7937 \end{aligned}$$

where again we computed the residues at the poles located at 0.5 and −0.8, respectively.

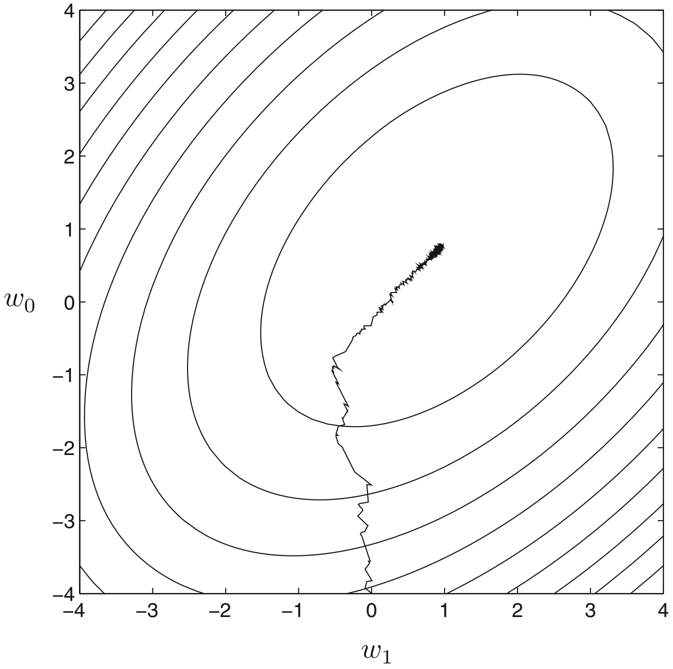


Fig. 3.4 Convergence path on the MSE surface

The correlation matrix of the adaptive-filter input signal is given by

$$\mathbf{R} = \begin{bmatrix} 1.6873 & -0.7937 \\ -0.7937 & 1.6873 \end{bmatrix}$$

(b) The coefficients corresponding to the Wiener solution are given by

$$\begin{aligned} \mathbf{w}_o &= \mathbf{R}^{-1}\mathbf{p} \\ &= 0.45106 \begin{bmatrix} 1.6873 & 0.7937 \\ 0.7937 & 1.6873 \end{bmatrix} \begin{bmatrix} 0.9524 \\ 0.4762 \end{bmatrix} \\ &= \begin{bmatrix} 0.8953 \\ 0.7034 \end{bmatrix} \end{aligned}$$

(c) The LMS algorithm is applied to minimize the MSE using a convergence factor $\mu = 1/40\text{tr}[\mathbf{R}]$, where $\text{tr}[\mathbf{R}] = 3.3746$. The value of μ is 0.0074. This small value of the convergence factor allows a smooth convergence path. The convergence path of the algorithm on the MSE surface is depicted in Fig. 3.4. As can be noted, the path followed by the LMS algorithm looks like a noisy steepest-descent path. It first approaches the main axis (eigenvector) corresponding to the smaller eigenvalue, and then follows toward the minimum in a direction increasingly aligned with this main axis.

(d) The learning curves of the MSE and the filter coefficients in a single run are depicted in Fig. 3.5. The learning curves of the MSE and the filter coefficients, obtained by averaging the results of 25 runs, are depicted in Fig. 3.6. As can be noted, these curves are less noisy than in the single run case. □

The adaptive-filtering problems discussed so far assumed that the signals taken from the environment were stochastic signals. Also, by assuming these signals were ergodic, we have shown that the adaptive filter is able to approach the Wiener solution by replacing the ensemble average by time averages. In conclusion, we can assume that the solution reached by the adaptive filter is based on time averages of the cross-correlations of the environment signals.

For example, if the environment signals are periodic deterministic signals, the optimal solution depends on the time average of the related cross-correlations computed over one period of the signals. Note that in this case, the solution obtained using an ensemble average would be time varying since we are dealing with a nonstationary problem. The following examples illustrate this issue.

Example 3.2. Suppose in an adaptive-filtering environment, the input signal consists of

$$x(k) = \cos(\omega_0 k)$$

The desired signal is given by

$$d(k) = \sin(\omega_0 k)$$

where $\omega_0 = \frac{2\pi}{M}$. In this case $M = 7$.

Compute the optimal solution for a first-order adaptive filter.

Solution. In this example, the signals involved are deterministic and periodic. If the adaptive-filter coefficients are fixed, the error is a periodic signal with period M. In this case, the objective function that will be minimized by the adaptive filter is the average value of the squared error defined by

$$\begin{aligned}\bar{E}[e^2(k)] &= \frac{1}{M}\sum_{m=0}^{M-1}\left[e^2(k-m)\right] \\ &= \bar{E}[d^2(k)] - 2\mathbf{w}^T\bar{\mathbf{p}} + \mathbf{w}^T\bar{\mathbf{R}}\mathbf{w} \end{aligned} \tag{3.80}$$

where

$$\bar{\mathbf{R}} = \begin{bmatrix} \bar{E}[\cos^2(\omega_0 k)] & \bar{E}[\cos(\omega_0 k)\cos(\omega_0(k-1))] \\ \bar{E}[\cos(\omega_0 k)\cos(\omega_0(k-1))] & \bar{E}[\cos^2(\omega_0 k)] \end{bmatrix}$$

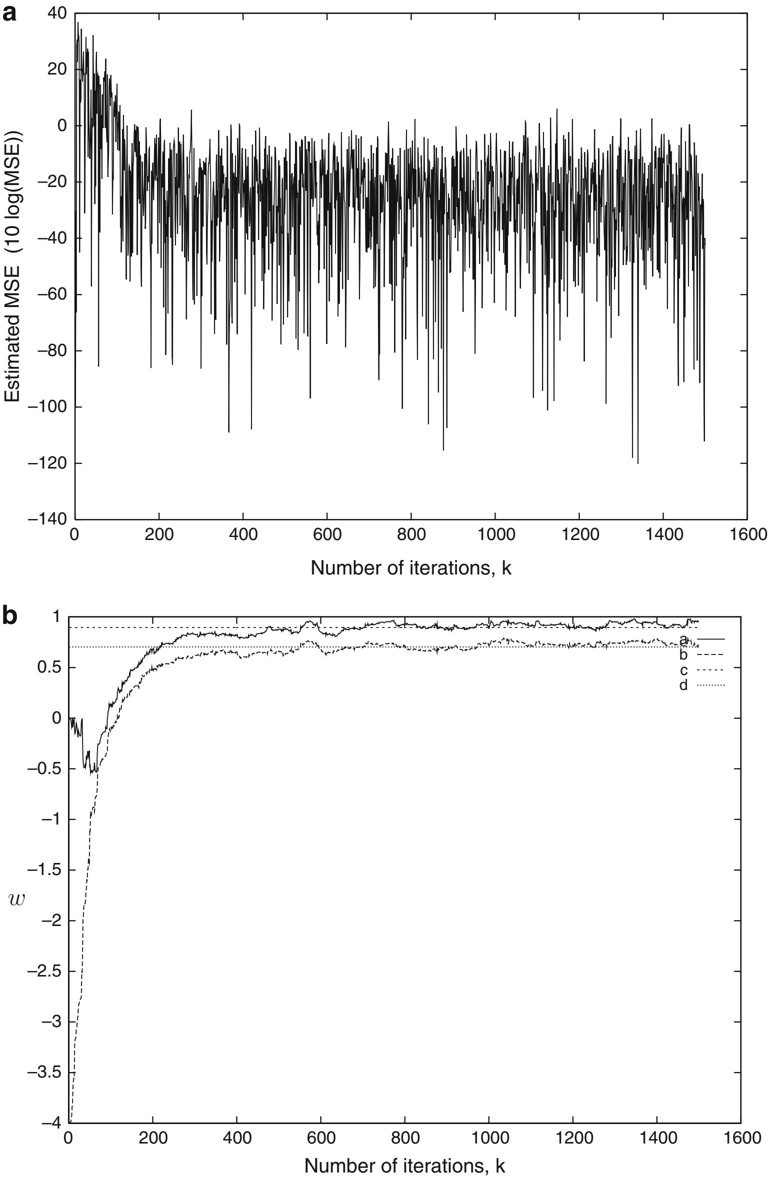


Fig. 3.5 (**a**) Learning curve of the instantaneous squared error (**b**) Learning curves of the coefficients, a—first coefficient, b—second coefficient, c—optimal value for the first coefficient, d—optimal value of the second coefficient

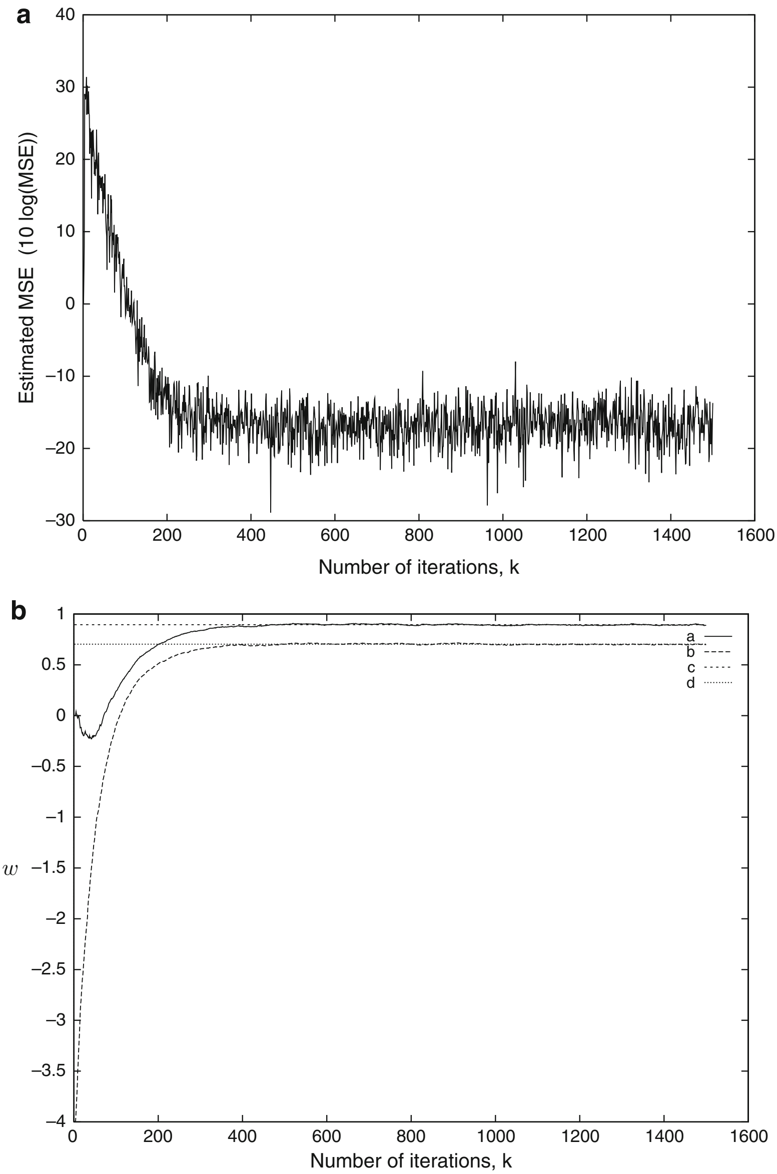


Fig. 3.6 (**a**) Learning curve of the MSE (**b**) Learning curves of the coefficients. Average of 25 runs. a—first coefficient, b—second coefficient, c—optimal value of the first coefficient, d—optimal value of the second coefficient

and

$$\bar{\mathbf{p}} = \left[\bar{E}[\sin(\omega_0 k)\cos(\omega_0 k)] \quad \bar{E}[\sin(\omega_0 k)\cos(\omega_0 k - 1)]\right]^T$$

The expression for the optimal coefficient vector can be easily derived.

$$\mathbf{w}_o = \bar{\mathbf{R}}^{-1}\bar{\mathbf{p}}$$

Now the above results are applied to the problem described. The elements of the vector $\bar{\mathbf{p}}$ are calculated as follows:

$$\begin{aligned}
\bar{\mathbf{p}} &= \frac{1}{M}\sum_{m=0}^{M-1}\begin{bmatrix} d(k-m)x(k-m) \\ d(k-m)x(k-m-1) \end{bmatrix} \\
&= \frac{1}{M}\sum_{m=0}^{M-1}\begin{bmatrix} \sin(\omega_0(k-m))\cos(\omega_0(k-m)) \\ \sin(\omega_0(k-m))\cos(\omega_0(k-m-1)) \end{bmatrix} \\
&= \frac{1}{2}\begin{bmatrix} 0 \\ \sin(\omega_0) \end{bmatrix} \\
&= \begin{bmatrix} 0 \\ 0.3909 \end{bmatrix}
\end{aligned}$$

The elements of the correlation matrix of the adaptive-filter input signal are calculated as follows:

$$\begin{aligned}
\bar{r}(i) &= \bar{E}[x(k)x(k-i)] \\
&= \frac{1}{M}\sum_{m=0}^{M-1}[\cos(\omega_0(k-m))\cos(\omega_0(k-m-i))]
\end{aligned}$$

where

$$\begin{aligned}
\bar{r}(0) &= \bar{E}[\cos^2(\omega_0(k))] = 0.5 \\
\bar{r}(1) &= \bar{E}[\cos(\omega_0(k))\cos(\omega_0(k-1))] = 0.3117
\end{aligned}$$

The correlation matrix of the adaptive-filter input signal is given by

$$\bar{\mathbf{R}} = \begin{bmatrix} 0.5 & 0.3117 \\ 0.3117 & 0.5 \end{bmatrix}$$

The coefficients corresponding to the optimal solution are given by

$$\bar{\mathbf{w}}_o = \bar{\mathbf{R}}^{-1}\bar{\mathbf{p}} = \begin{bmatrix} -0.7972 \\ 1.2788 \end{bmatrix} \qquad \square$$

Example 3.3. (a) Assume the input and desired signals are deterministic and periodic with period M. Study the LMS algorithm behavior.

(b) Choose an appropriate value for μ in the previous example and plot the convergence path for the LMS algorithm on the average error surface.

Solution. (a) It is convenient at this point to recall the coefficient updating of the LMS algorithm

$$\mathbf{w}(k+1) = \mathbf{w}(k) + 2\mu\mathbf{x}(k)e(k) = \mathbf{w}(k) + 2\mu\mathbf{x}(k)\left[d(k) - \mathbf{x}^T(k)\mathbf{w}(k)\right]$$

This equation can be rewritten as

$$\mathbf{w}(k+1) = \left[\mathbf{I} - 2\mu\mathbf{x}(k)\mathbf{x}^T(k)\right]\mathbf{w}(k) + 2\mu d(k)\mathbf{x}(k) \qquad (3.81)$$

The solution of (3.81), as a function of the initial values of the adaptive-filter coefficients, is given by

$$\mathbf{w}(k+1) = \prod_{i=0}^{k}\left[\mathbf{I} - 2\mu\mathbf{x}(i)\mathbf{x}^T(i)\right]\mathbf{w}(0) + \sum_{i=0}^{k}\left\{\prod_{j=i+1}^{k}\left[\mathbf{I} - 2\mu\mathbf{x}(j)\mathbf{x}^T(j)\right]2\mu d(i)\mathbf{x}(i)\right\} \qquad (3.82)$$

where we define that $\prod_{j=k+1}^{k}[\cdot] = 1$ for the second product.

Assuming the value of the convergence factor μ is small enough to guarantee that the LMS algorithm will converge, the first term on the right-hand side of the above equation will vanish as $k \to \infty$. The resulting expression for the coefficient vector is given by

$$\mathbf{w}(k+1) = \sum_{i=0}^{k}\left\{\prod_{j=i+1}^{k}\left[\mathbf{I} - 2\mu\mathbf{x}(j)\mathbf{x}^T(j)\right]2\mu d(i)\mathbf{x}(i)\right\}$$

The analysis of the above solution is not straightforward. Following an alternative path based on averaging the results in a period M, we can reach conclusive results.

Let us define the average value of the adaptive-filter parameters as follows:

$$\overline{\mathbf{w}(k+1)} = \frac{1}{M}\sum_{m=0}^{M-1}\mathbf{w}(k+1-m)$$

Similar definition can be applied to the remaining parameters of the algorithm.

Considering that the signals are deterministic and periodic, we can apply the average operation to (3.81). The resulting equation is

$$\begin{aligned}\overline{\mathbf{w}(k+1)} &= \frac{1}{M}\sum_{m=0}^{M-1}\left[\mathbf{I}-2\mu\mathbf{x}(k-m)\mathbf{x}^{T}(k-m)\right]\mathbf{w}(k-m)\\ &\quad +\frac{1}{M}\sum_{m=0}^{M-1}2\mu d(k-m)\mathbf{x}(k-m)\\ &= \overline{\left[\mathbf{I}-2\mu\mathbf{x}(k)\mathbf{x}^{T}(k)\right]\mathbf{w}(k)} + 2\mu\overline{d(k)\mathbf{x}(k)}\end{aligned} \tag{3.83}$$

For large k and small μ, it is expected that the parameters converge to the neighborhood of the optimal solution. In this case, we can consider that $\overline{\mathbf{w}(k+1)} \approx \overline{\mathbf{w}(k)}$ and that the following approximation is valid

$$\overline{\mathbf{x}(k)\mathbf{x}^{T}(k)\mathbf{w}(k)} \approx \overline{\mathbf{x}(k)\mathbf{x}^{T}(k)}\quad\overline{\mathbf{w}(k)}$$

since the parameters after convergence wander around the optimal solution. Using these approximations in (3.83), the average values of the parameters in the LMS algorithm for periodic signals are given by

$$\overline{\mathbf{w}(k)} \approx \overline{\mathbf{x}(k)\mathbf{x}^{T}(k)}^{-1}\,\overline{d(k)\mathbf{x}(k)} = \bar{\mathbf{R}}^{-1}\bar{\mathbf{p}}$$

(b) The LMS algorithm is applied to minimize the squared error of the problem described in Example 3.2 using a convergence factor $\mu = 1/100\mathrm{tr}[\bar{\mathbf{R}}]$, where $\mathrm{tr}[\bar{\mathbf{R}}] = 1$. The value of μ is 0.01. The convergence path of the algorithm on the MSE surface is depicted in Fig. 3.7. As can be verified, the parameters generated by the LMS algorithm approach the optimal solution. □

Example 3.4. The leaky LMS algorithm has the following updating equation

$$\mathbf{w}(k+1) = (1-2\mu\gamma)\mathbf{w}(k) + 2\mu e(k)\mathbf{x}(k) \tag{3.84}$$

where $0 < \gamma \ll 1$.

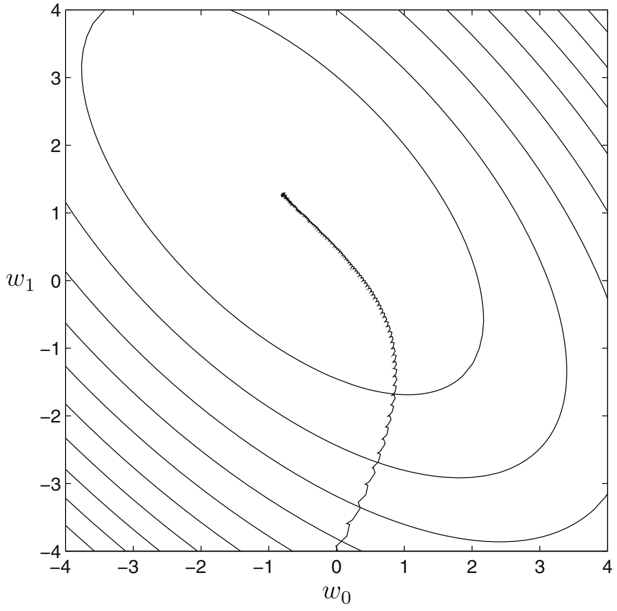


Fig. 3.7 Convergence path on the MSE surface

(a) Compute the range of values of μ such that the coefficients converge in average.
(b) What is the objective function this algorithm actually minimizes?
(c) What happens to the filter coefficients if the error and/or input signals become zero?

Solution. (a) By utilizing the error expression we generate the coefficient updating equation given by

$$\mathbf{w}(k+1) = \{\mathbf{I} - 2\mu[\mathbf{x}(k)\mathbf{x}^T(k) + \gamma\mathbf{I}]\}\mathbf{w}(k) + 2\mu d(k)\mathbf{x}(k)$$

By applying the expectation operation it follows that

$$E[\mathbf{w}(k+1)] = \{\mathbf{I} - 2\mu[\mathbf{R} + \gamma\mathbf{I}]\}E[\mathbf{w}(k)] + 2\mu\mathbf{p}$$

The inclusion of γ is equivalent to add a white noise to the input signal $x(n)$, such that a value of γ is added to the eigenvalues of the input signal autocorrelation matrix. As a result, the condition for the stability in the mean for the coefficients is expressed as

$$0 < \mu < \frac{1}{\lambda_{\max} + \gamma}$$

The coefficients converge to a biased solution with respect to the Wiener solution and are given by

$$E[\mathbf{w}(k)] = [\mathbf{R} + \gamma\mathbf{I}]^{-1}\mathbf{p}$$

for $k \to \infty$.

(b) Equation (3.84) can be rewritten in a form that helps us to recognize the gradient expression.

$$\begin{aligned}\mathbf{w}(k+1) &= \mathbf{w}(k) + 2\mu(-\gamma\mathbf{w}(k) + e(k)\mathbf{x}(k)) \\ &= \mathbf{w}(k) - 2\mu(\gamma\mathbf{w}(k) - d(k)\mathbf{x}(k) + \mathbf{x}(k)\mathbf{x}^T(k)\mathbf{w}(k)) \quad (3.85)\end{aligned}$$

By inspection we observe that in this case the gradient is described by

$$\mathbf{g}_\mathbf{w}(k) = 2\gamma\mathbf{w}(k) - 2e(k)\mathbf{x}(k) = 2\gamma\mathbf{w}(k) - 2d(k)\mathbf{x}(k) + 2\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{w}(k)$$

The corresponding objective function that is indeed minimized is given by

$$\xi(k) = \{\gamma||\mathbf{w}(k)||^2 + e^2(k)\}$$

(c) For zero input or zero error signal after some initial iterations, the dynamic updating (3.84) has zero excitation. Since the eigenvalues of the transition matrix $\{\mathbf{I} - 2\mu[\mathbf{x}(k)\mathbf{x}^T(k) + \gamma\mathbf{I}]\}$ are smaller than one, then the adaptive-filter coefficients will tend to zero for large k. □

3.6.2 System Identification Simulations

In this subsection, a system identification problem is described and solved by using the LMS algorithm. In the following chapters the same problem will be solved using other algorithms presented in the book. For the FIR adaptive filters the following identification problem is posed:

Example 3.5. An adaptive-filtering algorithm is used to identify a system with impulse response given below.

$$\mathbf{h} = [0.1\ 0.3\ 0.0\ -0.2\ -0.4\ -0.7\ -0.4\ -0.2]^T$$

Consider three cases for the input signal: colored noises with variance $\sigma_x^2 = 1$ and eigenvalue spread of their correlation matrix equal to 1.0, 20, and 80, respectively. The measurement noise is Gaussian white noise uncorrelated with the input and with variance $\sigma_n^2 = 10^{-4}$. The adaptive filter has eight coefficients.

(a) Run the algorithm and comment on the convergence behavior in each case.

(b) Measure the misadjustment in each example and compare with the theoretical results where appropriate.

(c) Considering that fixed-point arithmetic is used, run the algorithm for a set of experiments and calculate the expected values for $||\Delta\mathbf{w}(k)_Q||^2$ and $\xi(k)_Q$ for the following case:

Additional noise: white noise with variance	$\sigma_n^2 = 0.0015$
Coefficient wordlength:	$b_c = 16$ bits
Signal wordlength:	$b_d = 16$ bits
Input signal: Gaussian white noise with variance	$\sigma_x^2 = 1.0$

(d) Repeat the previous experiment for the following cases

$b_c = 12$ bits, $b_d = 12$ bits.
$b_c = 10$ bits, $b_d = 10$ bits.

(e) Suppose the unknown system is a time-varying system whose coefficients are first-order Markov processes with $\lambda_{\mathbf{w}} = 0.99$ and $\sigma_{\mathbf{w}}^2 = 0.0015$. The initial time-varying-system multiplier coefficients are the ones above described. The input signal is Gaussian white noise with variance $\sigma_x^2 = 1.0$, and the measurement noise is also Gaussian white noise independent of the input signal and of the elements of $\mathbf{n_w}(k)$, with variance $\sigma_n^2 = 0.01$. Simulate the experiment described, measure the total excess MSE, and compare to the calculated results.

Solution. (a) The colored input signal is generated by applying Gaussian white noise, with variance σ_v^2, to a first-order filter with transfer function

$$H(z) = \frac{z}{z-a}$$

As can be shown from (2.83), the input signal correlation matrix in this case is given by

$$\mathbf{R} = \frac{\sigma_v^2}{1-a^2}\begin{bmatrix} 1 & a & \cdots & a^7 \\ a & 1 & \cdots & a^6 \\ \vdots & \vdots & \ddots & \vdots \\ a^7 & a^6 & \cdots & 1 \end{bmatrix}$$

The proper choice of the value of a, in order to obtain the desired eigenvalue spread, is not a straightforward task. Some guidelines are now discussed. For example, if the adaptive filter is of first order, the matrix $\mathbf{R}$ is two by two with eigenvalues

$$\lambda_{\max} = \frac{\sigma_v^2}{1-a^2}(1+a)$$

and

$$\lambda_{\min} = \frac{\sigma_v^2}{1 - a^2}(1 - a)$$

respectively. In this case, the choice of a is straightforward.

In general, it can be shown that

$$\frac{\lambda_{\max}}{\lambda_{\min}} \leq \frac{|H_{\max}(e^{\jmath\omega})|^2}{|H_{\min}(e^{\jmath\omega})|^2}$$

For a very large order adaptive filter, the eigenvalue spread approaches

$$\frac{\lambda_{\max}}{\lambda_{\min}} \approx \frac{|H_{\max}(e^{\jmath\omega})|^2}{|H_{\min}(e^{\jmath\omega})|^2} = \left\{\frac{1 + a}{1 - a}\right\}^2$$

where the details to reach this result can be found in page 124 of [20].

Using the above relations as guidelines, we reached the correct values of a. These values are $a = 0.6894$ and $a = 0.8702$ for eigenvalue spreads of 20 and 80, respectively.

Since the variance of the input signal should be unity, the variance of the Gaussian white noise that produces $x(k)$ should be given by

$$\sigma_v^2 = 1 - a^2$$

For the LMS algorithm, we first calculate the upper bound for μ ($\mu_{\max}$) to guarantee the algorithm stability, and run the algorithm for $\mu_{\max}$, $\mu_{\max}/5$, and $\mu_{\max}/10$.

In this example, the LMS algorithm does not converge for $\mu = \mu_{\max} \approx 0.1$. The convergence behavior for $\mu_{\max}/5$ and $\mu_{\max}/10$ is illustrated through the learning curves depicted in Fig. 3.8, where in this case the eigenvalue spread is 1. Each curve is obtained by averaging the results of 200 independent runs. As can be noticed, the reduction of the convergence factor leads to a reduction in the convergence speed. Also note that for $\mu = 0.02$ the estimated MSE is plotted only for the first 400 iterations, enough to display the convergence behavior. In all examples the tap coefficients are initialized with zero. Fig. 3.9 illustrates the learning curves for the various eigenvalue spreads, where in each case the convergence factor is $\mu_{\max}/5$. As expected the convergence rate is reduced for a high eigenvalue spread.

(b) The misadjustment is measured and compared with the results obtained from the following relation

$$M = \frac{\mu(N + 1)\sigma_x^2}{1 - \mu(N + 1)\sigma_x^2}$$

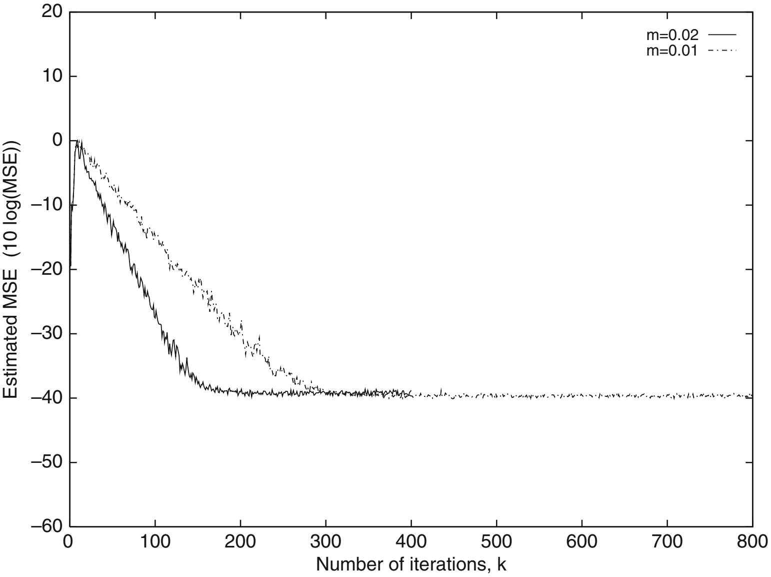


Fig. 3.8 Learning curves for the LMS algorithm with convergence factors $\mu_{\max}/5$ and $\mu_{\max}/10$

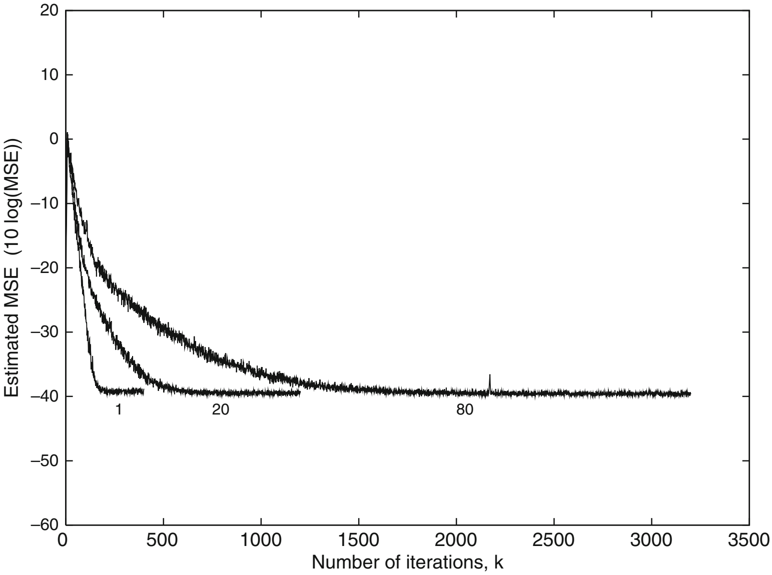


Fig. 3.9 Learning curves for the LMS algorithm for eigenvalue spreads: 1, 20, and 80

Table 3.1 Evaluation of the LMS algorithm

		Misadjustment		$\tau_{e_{\max}}$	$\tau_{w_{\max}}$	Iterations
μ	$\frac{\lambda_{\max}}{\lambda_{\min}}$	Experiment	Theory			
0.020000	1	0.2027	0.1905	12.5	25	58
0.012800	20	0.1298	0.1141	102.5	205	473
0.010240	80	0.1045	0.0892	338.9	677.5	1,561
0.010000	1	0.0881	0.0870	25	50	116
0.006401	20	0.0581	0.0540	205	410	944
0.005119	80	0.0495	0.0427	677.5	1,355	3,121

Also, for the present problem we calculated the time constants τ_{wi} and τ_{ei}, and the expected number of iterations to achieve convergence using the relations

$$\tau_{wi} \approx \frac{1}{2\mu\lambda_i}$$

$$\tau_{ei} \approx \frac{1}{4\mu\lambda_i}$$

$$k \approx \tau_{e_{\max}} \ln(100)$$

Table 3.1 illustrates the obtained results. As can be noted the analytical results agree with the experimental results, especially those related to the misadjustment. The analytical results related to the convergence time are optimistic as compared with the measured results. These discrepancies are mainly due to the approximations in the analysis.

(c), (d) The LMS algorithm is implemented employing fixed-point arithmetic using 16, 12, and 10 bits for data and coefficient wordlengths. The chosen value of μ is 0.01. The learning curves for the MSE are depicted in Fig. 3.10. Figure 3.11 depicts the evolution of $||\Delta\mathbf{w}(k)_Q||^2$ with the number of iterations. The experimental results show that the algorithm still works for such limited precision. In Table 3.2, we present a summary of the results obtained from simulation experiments and a comparison with the results predicted by the theory. The experimental results are obtained by averaging the results of 200 independent runs. The relations employed to calculate the theoretical results shown in Table 3.2 correspond to (15.26) and (15.32) derived in Chap. 15. These relations are repeated here for convenience:

$$E[||\Delta\mathbf{w}(k)_Q||^2] = \frac{\mu(\sigma_n^2 + \sigma_e^2)(N+1)}{1-\mu(N+1)\sigma_x^2} + \frac{(N+1)\sigma_{\mathbf{W}}^2}{4\mu\sigma_x^2[1-\mu(N+1)\sigma_x^2]}$$

$$\xi(k)_Q = \frac{\sigma_e^2 + \sigma_n^2}{1-\mu(N+1)\sigma_x^2} + \frac{(N+1)\sigma_{\mathbf{W}}^2}{4\mu[1-\mu(N+1)\sigma_x^2]}$$

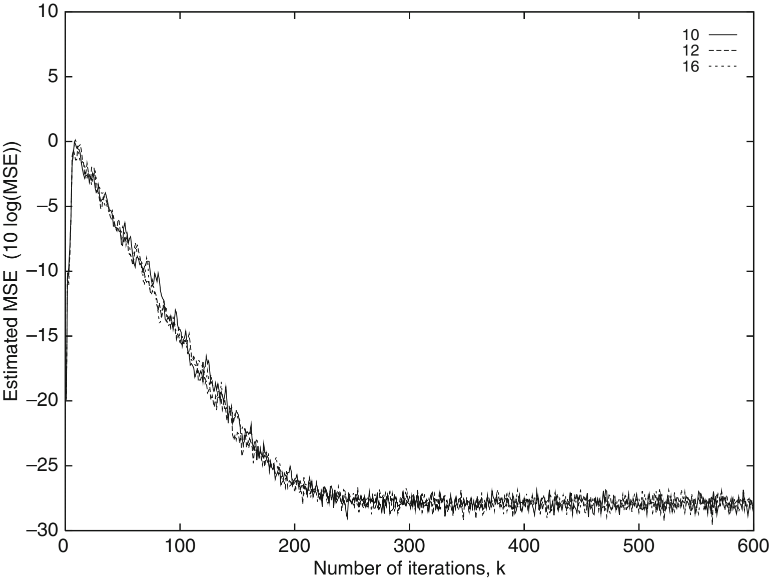


Fig. 3.10 Learning curves for the LMS algorithm implemented with fixed-point arithmetic and with $\mu = 0.01$

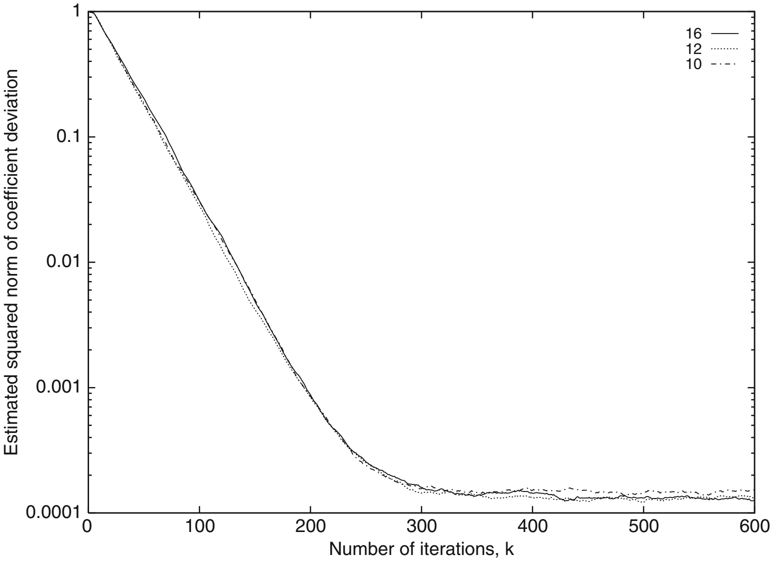


Fig. 3.11 Estimate of $||\Delta\mathbf{w}(k)_Q||^2$ for the LMS algorithm implemented with fixed-point arithmetic and with $\mu = 0.01$

Table 3.2 Results of the finite precision implementation of the LMS algorithm

	$\xi(k)_Q$		$E[\|\|\Delta\mathbf{w}(k)_Q\|\|^2]$	
No. of bits	Experiment	Theory	Experiment	Theory
16	$1.629\ 10^{-3}$	$1.630\ 10^{-3}$	$1.316\ 10^{-4}$	$1.304\ 10^{-4}$
12	$1.632\ 10^{-3}$	$1.631\ 10^{-3}$	$1.309\ 10^{-4}$	$1.315\ 10^{-4}$
10	$1.663\ 10^{-3}$	$1.648\ 10^{-3}$	$1.465\ 10^{-4}$	$1.477\ 10^{-4}$

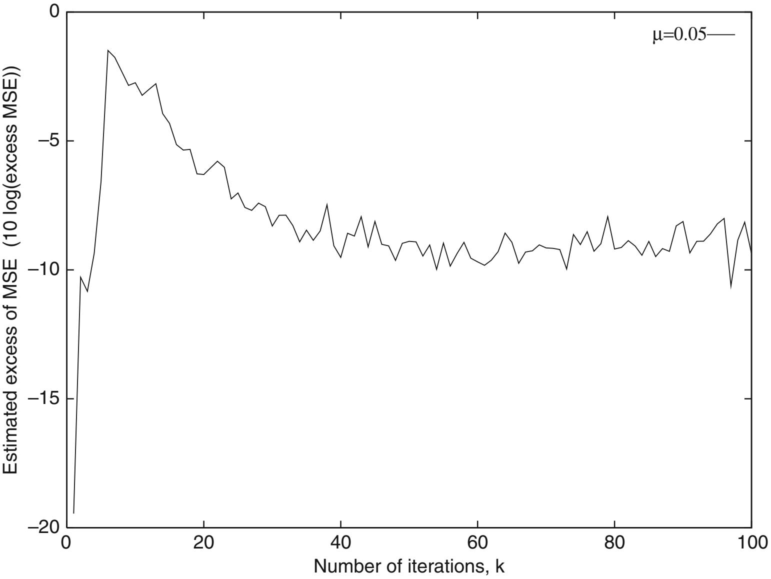


Fig. 3.12 The excess MSE of the LMS algorithm in nonstationary environment, $\mu = 0.05$

The results of Table 3.2 confirm that the finite-precision implementation analysis presented is accurate.

(e) The performance of the LMS algorithm is also tested in the nonstationary environment above described. The excess MSE is measured and depicted in Fig. 3.12. For this example μ_{opt} is found to be greater than $\mu_{\max}$. The value of μ used in the example is 0.05. The excess MSE in steady state predicted by the relation

$$\xi_{\text{total}} \approx \frac{\mu\sigma_n^2\text{tr}[\mathbf{R}]}{1-\mu\text{tr}[\mathbf{R}]} + \frac{\sigma_{\mathbf{w}}^2}{4\mu}\sum_{i=0}^{N}\frac{1}{1-\mu\lambda_i}$$

is 0.124, whereas the measured excess MSE in steady state is 0.118. Once more the results obtained from the analysis are accurate. □

3.6.3 Channel Equalization Simulations

In this subsection an equalization example is described. This example will be used as pattern for comparison of several algorithms presented in this book.

Example 3.6. Perform the equalization of a channel with the following impulse response

$$h(k) = 0.1\ (0.5^k)$$

for $k = 0, 1, \ldots 8$. Use a known training signal that consists of independent binary samples (−1,1). An additional Gaussian white noise with variance $10^{-2.5}$ is present at the channel output.

(a) Find the impulse response of an equalizer with 50 coefficients.
(b) Convolve the equalizer impulse response at a given iteration after convergence, with the channel impulse response and comment on the result.

Solution. (a) We apply the LMS algorithm to solve the equalization problem. We use $\mu_{\max}/5$ for the value of the convergence factor. In order to obtain $\mu_{\max}$, the values of $\lambda_{\max} = 0.04275$ and $\sigma_x^2 = 0.01650$ are measured and applied in (3.30). The resulting value of μ is 0.2197.
(b) The appropriate value of L is found to be round $(\frac{9+50}{2}) = 30$. The impulse response of the resulting equalizer is shown in Fig. 3.13. By convolving this response with the channel impulse response, we obtain the result depicted in Fig. 3.14 that clearly approximates an impulse. The measured MSE is 0.3492.

□

3.6.4 Fast Adaptation Simulations

The exact evaluation of the learning curves of the squared error or coefficients of an adaptive filter is a difficult task. In general the solution is to run repeated simulations and average their results. For the LMS algorithm this ensemble averaging leads to results which are close to those predicted by independence theory [4], if the convergence factor is small. In fact, the independence theory is a first-order approximation in μ to the actual learning curves of $\xi(k)$ [4, 21].

However, for large μ the results from the ensemble average can be quite different from the theoretical prediction [22]. The following example explores this observation.

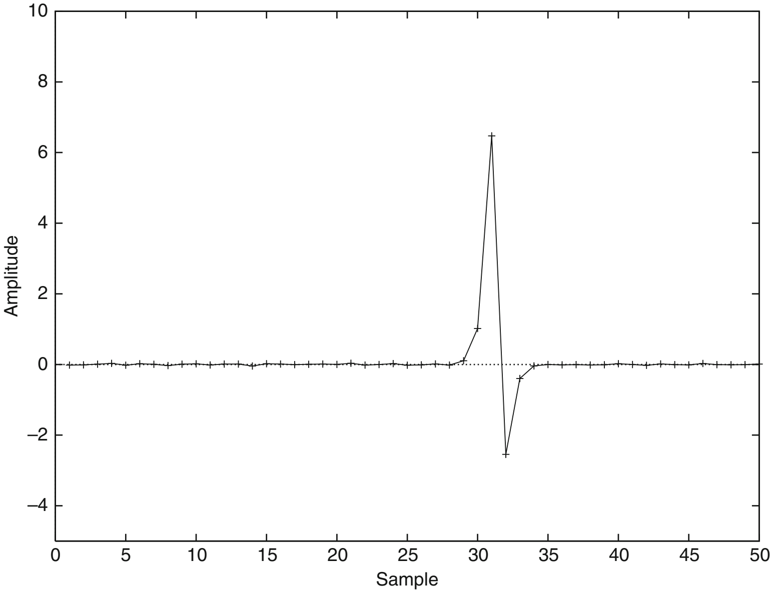


Fig. 3.13 Equalizer impulse response; LMS algorithm

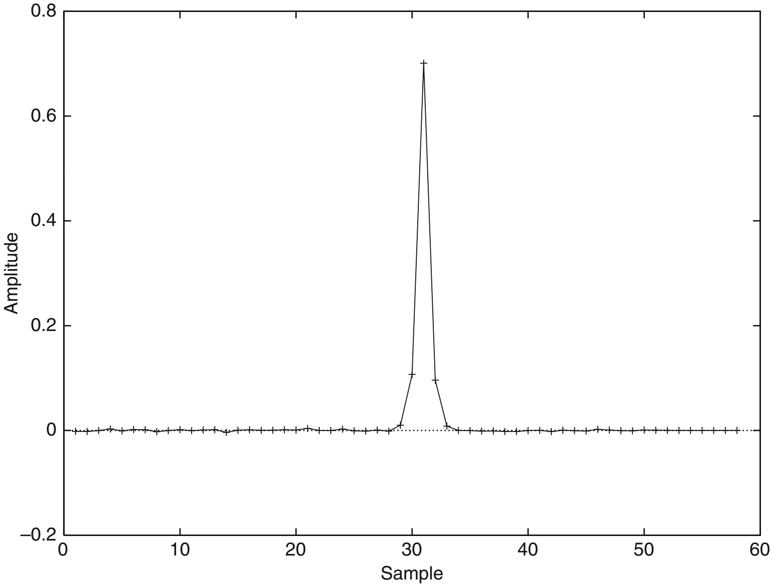


Fig. 3.14 Convolution result; LMS algorithm

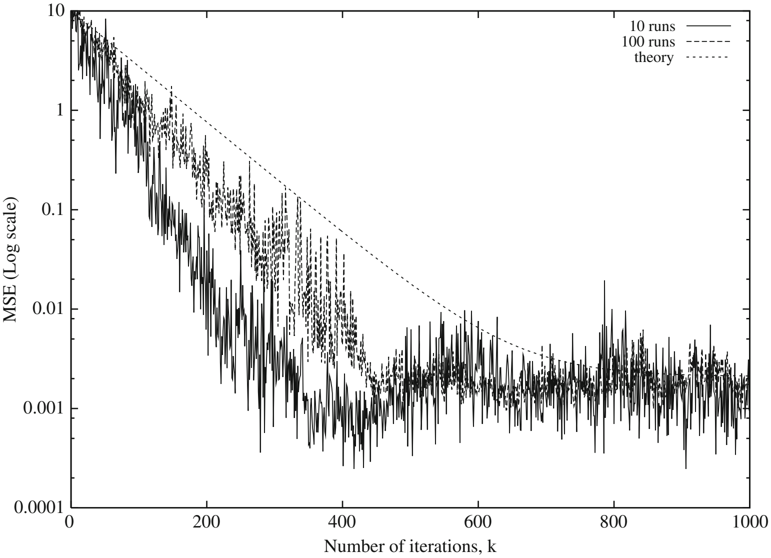


Fig. 3.15 Learning curves for the LMS algorithm with convergence factor $\mu = 0.08$, result of ensemble averages with 10 and 100 independent simulations as well as the theoretical curve

Example 3.7. An adaptive-filtering algorithm is used to identify a system. Consider three cases described below.

(a) The unknown system has length 10, the input signal is a stationary Gaussian noise with variance $\sigma_x^2 = 1$ and the measurement noise is Gaussian white noise uncorrelated with the input and with variance $\sigma_n^2 = 10^{-4}$.
(b) The unknown system has length 2, the input signal is a stationary uniformly distributed noise in the range -0.5 and 0.5, and there is no measurement noise.
(c) Study the behavior of the ensemble average as well as the mean square value of the coefficient error of an LMS algorithm with a single coefficient, when the input signal is a stationary uniformly distributed noise in the range $-a$ and a, and there is no measurement noise.

Solution. (a) Figure 3.15 depicts the theoretical learning curve for the squared error obtained using the independence theory as well as the curves obtained by averaging the results of 10 and 100 independent runs. The chosen convergence factor is $\mu = 0.08$. As we can observe the simulation curves are not close to the theoretical one, but they get closer as the number of independent runs increases.

(b) Figure 3.16 shows the exact theoretical learning curve for the squared error obtained from [23] along with the curves obtained by averaging the results of 100, 1,000 and 10,000 independent runs. The chosen convergence factor is $\mu = 4.00$. As we can observe the theoretical learning curve diverges whereas the simulation curves converge. A closer look at this problem is given in the next item.

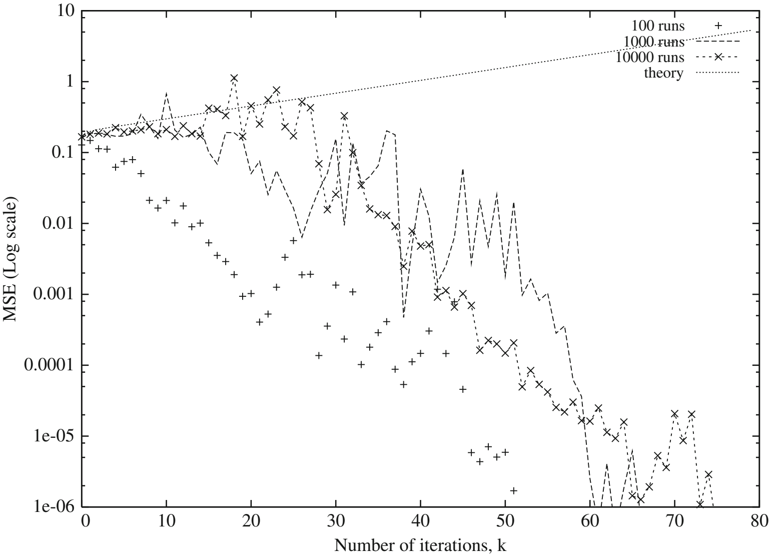


Fig. 3.16 Learning curves for the LMS algorithm with convergence factor $\mu = 4.00$, result of ensemble averages with 100, 1,000 and 10,000 independent simulations as well as the theoretical curve

(c) From (3.12), the evolution of the squared deviation in the tap coefficient is given by

$$\Delta w^2(k+1) = \left[1 - 2\mu x^2(k)\right]^2 \Delta w^2(k)$$

where $\Delta w(0)$ is fixed, and the additional noise is zero. Note that the evolution of $\Delta w^2(k)$ is governed by the random factor $\left[1 - 2\mu x^2(k)\right]^2$. With the assumptions on the input signal these random factors form an independent, identically distributed random sequence. The above model can then be rewritten as

$$\Delta w^2(k+1) = \left\{\prod_{i=0}^{k}\left[1 - 2\mu x^2(i)\right]^2\right\} \Delta w^2(0) \tag{3.86}$$

The objective now is to study the differences between the expected value of $\Delta w^2(k+1)$ and its ensemble average. In the first case, by using the independence of the random factors in (3.86) we have that

$$\begin{aligned} E[\Delta w^2(k+1)] &= \left\{\prod_{i=0}^{k} E\left[(1 - 2\mu x^2(i))^2\right]\right\} \Delta w^2(0) \\ &= \left\{E\left[(1 - 2\mu x^2(0))^2\right]\right\}^{k+1} \Delta w^2(0) \end{aligned} \tag{3.87}$$

Since the variance of the input signal is $\sigma_x^2 = \frac{a^2}{3}$ and its fourth-order moment is given by $\frac{a^4}{5}$, the above equation can be rewritten as

$$E[\Delta w^2(k+1)] = \left\{E\left[(1-2\mu x^2(0))^2\right]\right\}^{k+1} \Delta w^2(0)$$
$$= \left(1 - 4\mu\frac{a^2}{3} + 4\mu^2\frac{a^4}{5}\right)^{k+1} \Delta w^2(0) \tag{3.88}$$

From the above equation we can observe that the rate of convergence of $E[\Delta w^2(k)]$ is equal to $\ln\{E\left[(1-2\mu x^2(0))^2\right]\}$.

Let's examine now how the ensemble average of $\Delta w^2(k)$ evolves, for large k and μ, by computing its logarithm as follows:

$$\ln[\Delta w^2(k+1)] = \sum_{i=0}^{k} \ln[(1-2\mu x^2(i))^2] + \ln[\Delta w^2(0)] \tag{3.89}$$

By assuming that $\ln[(1-2\mu x^2(i))^2]$ exists and by employing the law of large numbers [13], we obtain

$$\frac{\ln[\Delta w^2(k+1)]}{k+1} = \frac{1}{k+1}\left\{\sum_{i=0}^{k} \ln[(1-2\mu x^2(i))^2] + \ln[\Delta w^2(0)]\right\} \tag{3.90}$$

which converges asymptotically to

$$E\left\{\ln\left[(1-2\mu x^2(i))^2\right]\right\}$$

For large k, after some details found in [22], from the above relation it can be concluded that

$$\Delta w^2(k+1) \approx C\mathrm{e}^{(k+1)E\{\ln[(1-2\mu x^2(i))^2]\}} \tag{3.91}$$

where C is a positive number which is not a constant and will be different for each run of the algorithm. In fact, C can have quite large values for some particular runs. In conclusion, the ensemble average of $\Delta w^2(k+1)$ decreases or increases with a time constant close to $E\{\ln[(1-2\mu x^2(i))^2]\}^{-1}$. Also it converges to zero if and only if $E\{\ln[(1-2\mu x^2(i))^2]\} < 0$, leading to a distinct convergence condition on $2\mu x^2(i)$ from that obtained by the mean-square stability. In fact, there is a range of values of the convergence factor in which the ensemble average converges but the mean-square value diverges, explaining the convergence behavior in Fig. 3.16.

Figure 3.17 depicts the curves of $\ln\{E\left[(1-2\mu x^2(0))^2\right]\}$ (the logarithm of the rate of convergence of mean-square coefficient error) and of $E\{\ln[(1-2\mu x^2(i))^2]\}$ as a function of $2\mu x^2(i)$. For small values of $2\mu x^2(i)$ both curves

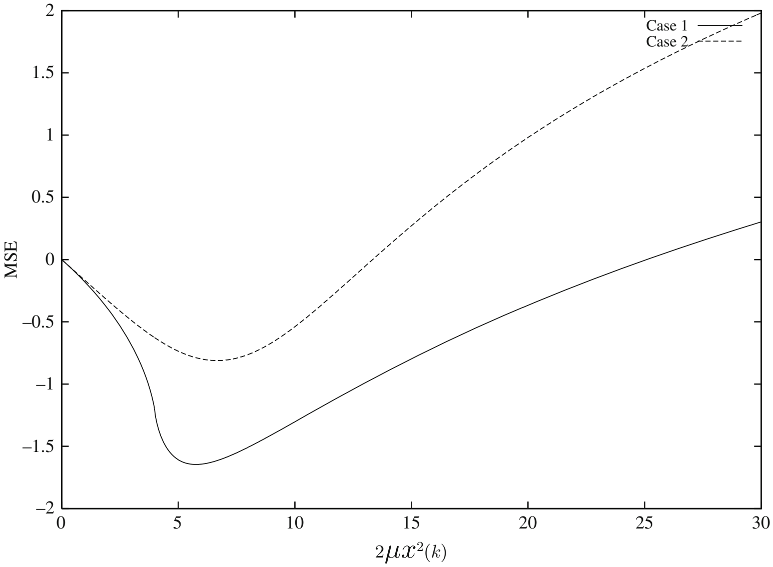


Fig. 3.17 Parameters related to the rate of convergence, Case 1: $E\{[\ln[(1-2\mu x^2(i))^2]\}$, Case 2: $\ln\{E\left[(1-2\mu x^2(0))^2\right]\}$ as a function of $2\mu x^2(i)$

are quite close, however for larger values they are somewhat different in particular at the minima of the curves which correspond to the fastest convergence rate. In addition, as the curves become further apart the convergence is faster for the ensemble average of the squared coefficient error than for the mean-square coefficient error for large k. □

3.6.5 *The Linearly Constrained LMS Algorithm*

In the narrowband beamformer application discussed in Sect. 2.5, our objective was to minimize the array output power subjecting the linear combiner coefficients to a set of constraints. Now, let us derive an adaptive version of the LCMV filter by first rewriting the linearly constrained objective function of (2.107) for the case of multiple constraints as

$$\begin{aligned}\xi_c &= E\left[\mathbf{w}^T\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{w}\right] + \boldsymbol{\lambda}^T\left[\mathbf{C}^T\mathbf{w} - \mathbf{f}\right]\\ &= \mathbf{w}^T\mathbf{R}\mathbf{w} + \boldsymbol{\lambda}^T\left[\mathbf{C}^T\mathbf{w} - \mathbf{f}\right] \qquad (3.92)\end{aligned}$$

where $\mathbf{R}$ is the input signal autocorrelation matrix, $\mathbf{C}$ is the constraint matrix, and $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers.

The constrained LMS-based algorithm [24] can be derived by searching for the coefficient vector $\mathbf{w}(k+1)$ that satisfies the set of constraints and represents a small update with respect to $\mathbf{w}(k)$ in the direction of the negative of the gradient (see (2.108)), i.e.,

$$\begin{aligned}\mathbf{w}(k+1) &= \mathbf{w}(k) - \mu \mathbf{g}_{\mathbf{w}}\{\xi_c(k)\} \\ &= \mathbf{w}(k) - \mu[2\mathbf{R}(k)\mathbf{w}(k) + \mathbf{C}\boldsymbol{\lambda}(k)]\end{aligned} \tag{3.93}$$

where $\mathbf{R}(k)$ is some estimate of the input signal autocorrelation matrix at instant k, $\mathbf{C}$ is again the constraint matrix, and $\boldsymbol{\lambda}(k)$ is the $(N+1) \times 1$ vector of Lagrange multipliers.

In the particular case of the constrained LMS algorithm, matrix $\mathbf{R}(k)$ is chosen as an instantaneous rank-one estimate given by $\mathbf{x}(k)\mathbf{x}^T(k)$. In this case, we can utilize the method of Lagrange multipliers to solve the constrained minimization problem defined by

$$\begin{aligned}\xi_c(k) &= \mathbf{w}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{w}(k) + \boldsymbol{\lambda}^T(k)\left[\mathbf{C}^T\mathbf{w}(k) - \mathbf{f}\right] \\ &= \mathbf{w}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{w}(k) + \left[\mathbf{w}^T(k)\mathbf{C} - \mathbf{f}^T\right]\boldsymbol{\lambda}(k)\end{aligned} \tag{3.94}$$

The gradient of $\xi_c(k)$ with respect to $\mathbf{w}(k)$ is given by

$$\mathbf{g}_{\mathbf{w}}\{\xi_c(k)\} = 2\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{w}(k) + \mathbf{C}\boldsymbol{\lambda}(k) \tag{3.95}$$

The constrained LMS updating algorithm related to (3.93) becomes

$$\begin{aligned}\mathbf{w}(k+1) &= \mathbf{w}(k) - 2\mu\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{w}(k) - \mu\mathbf{C}\boldsymbol{\lambda}(k) \\ &= \mathbf{w}(k) - 2\mu y(k)\mathbf{x}(k) - \mu\mathbf{C}\boldsymbol{\lambda}(k)\end{aligned} \tag{3.96}$$

If we apply the constraint relation $\mathbf{C}^T\mathbf{w}(k+1) = \mathbf{f}$ to the above expression, it follows that

$$\begin{aligned}\mathbf{C}^T\mathbf{w}(k+1) &= \mathbf{f} \\ &= \mathbf{C}^T\mathbf{w}(k) - 2\mu\mathbf{C}^T\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{w}(k) - \mu\mathbf{C}^T\mathbf{C}\boldsymbol{\lambda}(k) \\ &= \mathbf{C}^T\mathbf{w}(k) - 2\mu y(k)\mathbf{C}^T\mathbf{x}(k) - \mu\mathbf{C}^T\mathbf{C}\boldsymbol{\lambda}(k)\end{aligned} \tag{3.97}$$

By solving the above equation for $\mu\boldsymbol{\lambda}(k)$ we get

$$\mu\boldsymbol{\lambda}(k) = \left[\mathbf{C}^T\mathbf{C}\right]^{-1}\mathbf{C}^T\left[\mathbf{w}(k) - 2\mu y(k)\mathbf{x}(k)\right] - \left[\mathbf{C}^T\mathbf{C}\right]^{-1}\mathbf{f} \tag{3.98}$$

If we substitute (3.98) in the updating (3.96), we obtain

$$\mathbf{w}(k+1) = \mathbf{P}[\mathbf{w}(k) - 2\mu y(k)\mathbf{x}(k)] + \mathbf{f}_c \tag{3.99}$$

where $\mathbf{f}_c = \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{f}$ and $\mathbf{P} = \mathbf{I} - \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T$. Notice that the updated coefficient vector given in (3.99) is a projection onto the hyperplane defined by $\mathbf{C}^T\mathbf{w} = \mathbf{0}$ of an unconstrained LMS solution plus a vector $\mathbf{f}_c$ that brings the projected solution back to the constraint hyperplane.

If there is a reference signal $d(k)$, the updating equation is given by

$$\mathbf{w}(k+1) = \mathbf{P}\mathbf{w}(k) + 2\mu e(k)\mathbf{P}\mathbf{x}(k) + \mathbf{f}_c \tag{3.100}$$

In the case of the constrained normalized LMS algorithm (see Sect. 4.4), the solution satisfies $\mathbf{w}^T(k+1)\mathbf{x}(k) = d(k)$ in addition to $\mathbf{C}^T\mathbf{w}(k+1) = \mathbf{f}$ [25]. Alternative adaptation algorithms may be derived such that the solution at each iteration also satisfies a set of linear constraints [26].

For environments with complex signals and complex constraints, the updating equation is given by

$$\mathbf{w}(k+1) = \mathbf{P}\mathbf{w}(k) + \mu_c e^*(k)\mathbf{P}\mathbf{x}(k) + \mathbf{f}_c \tag{3.101}$$

where $\mathbf{C}^H\mathbf{w}(k+1) = \mathbf{f}$, $\mathbf{f}_c = \mathbf{C}(\mathbf{C}^H\mathbf{C})^{-1}\mathbf{f}$ and $\mathbf{P} = \mathbf{I} - \mathbf{C}(\mathbf{C}^H\mathbf{C})^{-1}\mathbf{C}^H$.

An efficient implementation for constrained adaptive filters was proposed in [27], which consists of applying a transformation to the input signal vector based on Householder transformation. The method can be regarded as an alternative implementation of the generalized sidelobe canceller structure, but with the advantages of always utilizing orthogonal/unitary matrices and rendering low computational complexity.

Example 3.8. An array of antennas with four elements, with inter-element spacing of 0.15 m, receives signals from two different sources arriving at 90° and 30° of angles with respect to the axis where the antennas are placed. The desired signal impinges on the antenna at 90°. The signal of interest is a sinusoid of frequency 20 MHz and the interferer signal is a sinusoid of frequency 70 MHz. The sampling frequency is 2 GHz.

Use the linearly constrained LMS algorithm in order to adapt the array coefficients.

Solution. The adaptive-filter coefficients are initialized with $\mathbf{w}(0) = \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{f}$. The value of μ used is 0.1. Figure 3.18 illustrates the learning curve for the output signal. Figure 3.19 illustrates details of the output signal in the early iterations where we can observe the presence of both sinusoid signals. In Fig. 3.20, the details of the output signal after convergence shows that mainly the desired sinusoid signal is present. The array output power response after convergence, as a function of the angle of arrival, is depicted in Fig. 3.21. From this figure, we observe the attenuation imposed by the array on signals arriving at 30° of angle, where the interference signal impinges. □

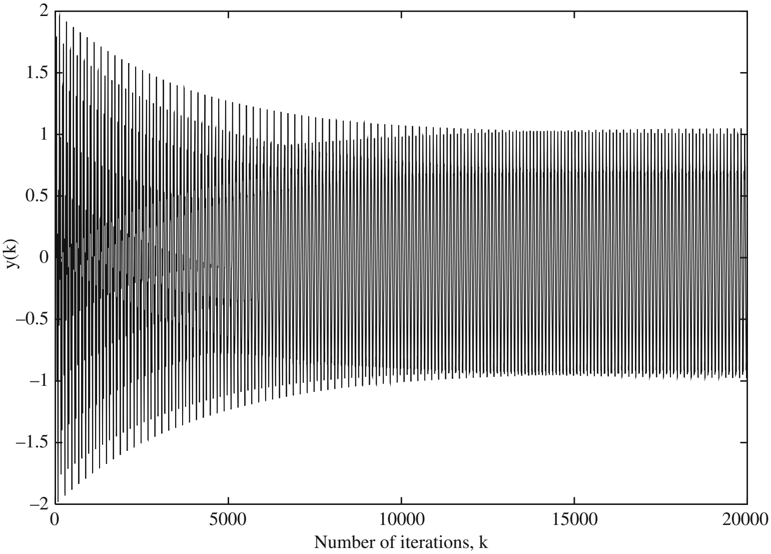
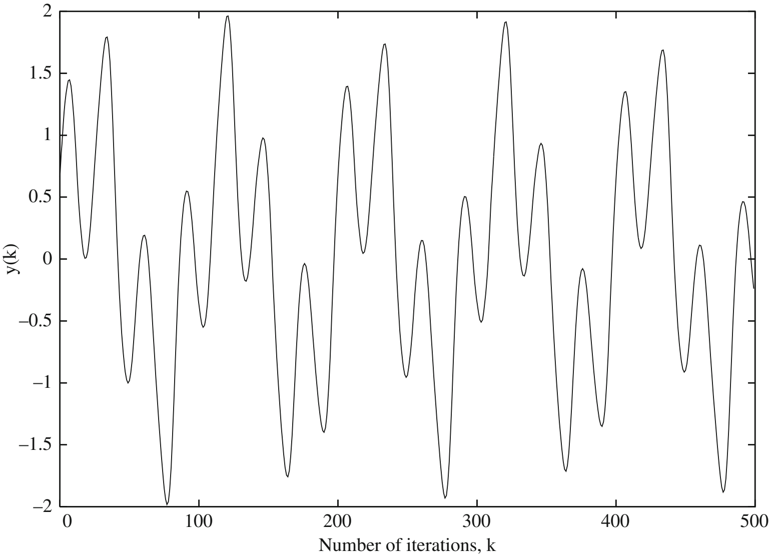


Fig. 3.18 Learning curves for the linearly constrained LMS algorithm with convergence factor $\mu = 0.1$



(a) Early output signal.

Fig. 3.19 Learning curves for the linearly constrained LMS algorithm with convergence factor $\mu = 0.1$; early output signal

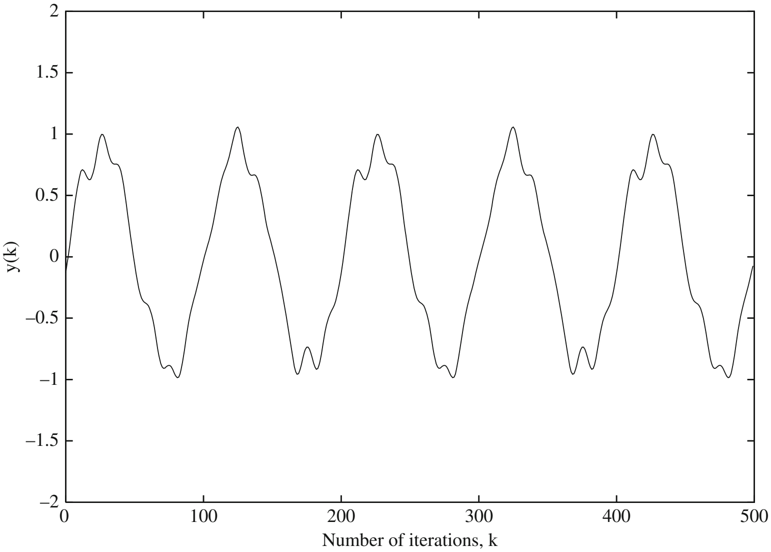


Fig. 3.20 Learning curves for the linearly constrained LMS algorithm with convergence factor $\mu = 0.1$; output signal after convergence

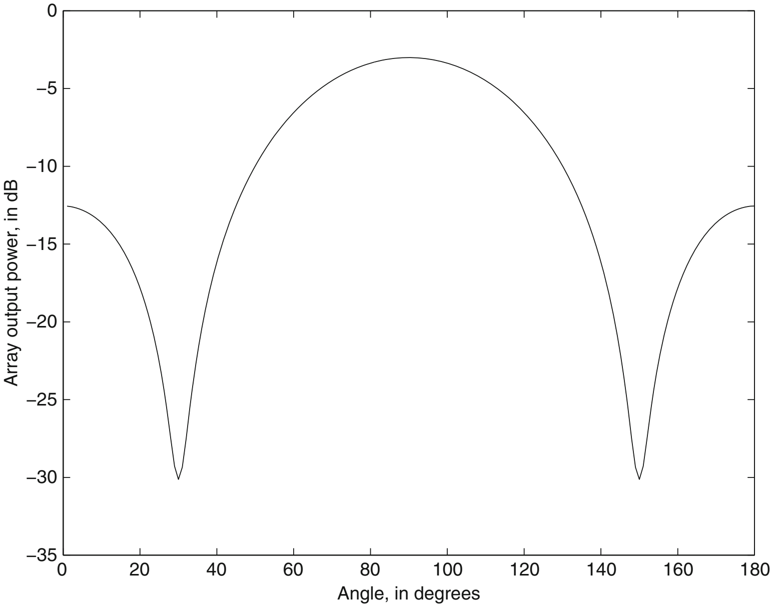


Fig. 3.21 Array output power after convergence, as a function of the angle of arrival

3.7 Concluding Remarks

In this chapter, we studied the LMS adaptive algorithm that is certainly the most popular among the adaptive-filtering algorithms. The attractiveness of the LMS algorithm is due to its simplicity and accessible analysis under idealized conditions. As demonstrated in the present chapter, the noisy estimate of the gradient that is used in the LMS algorithm is the main source of loss in performance for stationary environments. Further discussions on the convergence behavior and on the optimality of the LMS algorithm have been reported in the open literature, see for example [28–34].

For nonstationary environments we showed how the algorithm behaves assuming the optimal parameter can be modeled as a first-order Markov process. The analysis allowed us to determine the conditions for adequate tracking and acceptable excess MSE. Further analysis can be found in [35].

The quantization effects on the behavior of the LMS algorithm are presented in Chap. 15. The algorithm is fairly robust against quantization errors, and this is for sure one of the reasons for its choice in a number of practical applications [36, 37].

A number of simulation examples with the LMS algorithm was presented in this chapter. The simulations included examples in system identification and equalization. Also a number of theoretical results derived in the present chapter were verified, such as the excess MSE in stationary and nonstationary environments, the finite-precision analysis, etc.

3.8 Problems

1. The LMS algorithm is used to predict the signal $x(k) = \cos(\pi k/3)$ using a second-order FIR filter with the first tap fixed at 1, by minimizing the mean squared value of $y(k)$. Calculate an appropriate μ, the output signal, and the filter coefficients for the first ten iterations. Start with $\mathbf{w}^T(0) = [1\ 0\ 0]$.
2. The signal

$$x(k) = -0.85x(k-1) + n(k)$$

   is applied to a first-order predictor, where $n(k)$ is Gaussian white noise with variance $\sigma_n^2 = 0.3$.

   (a) Compute the Wiener solution.
   (b) Choose an appropriate value for μ and plot the convergence path for the LMS algorithm on the MSE error surface.
   (c) Plot the learning curves for the MSE and the filter coefficients in a single run as well as for the average of 25 runs.

3. Assuming it is desired to minimize the objective function $E[e^4(k)]$ utilizing a stochastic gradient type of algorithm such as the LMS. The resulting algorithm is called least-mean fourth algorithm [38]. Derive this algorithm.

4. The data-reusing LMS algorithm has the following updating equation

$$\begin{aligned}\hat{e}_l(k) &= d(k) - \hat{\mathbf{w}}_l^T(k)\mathbf{x}(k)\\ \hat{\mathbf{w}}_{l+1}(k) &= \hat{\mathbf{w}}_l(k) + 2\mu\hat{e}_l(k)\mathbf{x}(k)\end{aligned} \tag{3.102}$$

for $l = 0, 1, \ldots, L-1$, and

$$\mathbf{w}(k+1) = \hat{\mathbf{w}}_L(k) = \hat{\mathbf{w}}_{L-1}(k) + 2\mu\hat{e}_{L-1}(k)\mathbf{x}(k) \tag{3.103}$$

where $\hat{\mathbf{w}}_0(k) = \mathbf{w}(k)$.

(a) Compute the range of values of μ such that the coefficients converge in average.
(b) What is the objective function this algorithm actually minimizes?
(c) Compare its convergence speed and computational complexity with the LMS algorithm.

5. The momentum LMS algorithm has the following updating equation

$$\mathbf{w}(k+1) = \mathbf{w}(k) + 2\mu e(k)\mathbf{x}(k) + \gamma[\mathbf{w}(k) - \mathbf{w}(k-1)] \tag{3.104}$$

for $|\gamma| < 1$.

(a) Compute the range of values of μ such that the coefficients converge in average.
(b) What is the objective function this algorithm actually minimizes?
(c) Show that this algorithm can have faster convergence and higher misadjustment than the LMS algorithm.

6. An LMS algorithm can be updated in a block form. For a block of length 2 the updating equations have the following form.

$$\begin{aligned}\begin{bmatrix} e(k)\\ e(k-1)\end{bmatrix} &= \begin{bmatrix} d(k)\\ d(k-1)\end{bmatrix} - \begin{bmatrix}\mathbf{x}^T(k)\mathbf{w}(k)\\ \mathbf{x}^T(k-1)\mathbf{w}(k-1)\end{bmatrix}\\ &= \begin{bmatrix} d(k)\\ d(k-1)\end{bmatrix} - \begin{bmatrix}\mathbf{x}^T(k)\\ \mathbf{x}^T(k-1)\end{bmatrix}\mathbf{w}(k-1)\\ &\quad - \begin{bmatrix} 0 & 2\mu\mathbf{x}^T(k)\mathbf{x}(k-1)\\ 0 & 0\end{bmatrix}\begin{bmatrix} e(k)\\ e(k-1)\end{bmatrix}\end{aligned}$$

This relation, in a more compact way, is equivalent to

$$\begin{bmatrix} e(k)\\ e(k-1)\end{bmatrix} = \begin{bmatrix} 1 & -2\mu\mathbf{x}^T(k)\mathbf{x}(k-1)\\ 0 & 1\end{bmatrix}^{-1}\left\{\begin{bmatrix} d(k)\\ d(k-1)\end{bmatrix} - \begin{bmatrix}\mathbf{x}^T(k)\\ \mathbf{x}^T(k-1)\end{bmatrix}\mathbf{w}(k-1)\right\} \tag{3.105}$$

Derive an expression for a block of length $L+1$.

7. Use the LMS algorithm to identify a system with the transfer function given below. The input signal is a uniformly distributed white noise with variance $\sigma_x^2 = 1$, and the measurement noise is Gaussian white noise uncorrelated with the input with variance $\sigma_n^2 = 10^{-3}$. The adaptive filter has 12 coefficients.

$$H(z) = \frac{1 - z^{-12}}{1 - z^{-1}}$$

   (a) Calculate the upper bound for μ ($\mu_{\max}$) to guarantee the algorithm stability.
   (b) Run the algorithm for $\mu_{\max}/2$, $\mu_{\max}/10$, and $\mu_{\max}/50$. Comment on the convergence behavior in each case.
   (c) Measure the misadjustment in each example and compare with the results obtained by (3.50).
   (d) Plot the obtained FIR filter frequency response at any iteration after convergence is achieved and compare with the unknown system.

8. Repeat the previous problem using an adaptive filter with eight coefficients and interpret the results.
9. Repeat problem 2 in case the input signal is a uniformly distributed white noise with variance $\sigma_{n_x}^2 = 0.5$ filtered by an all-pole filter given by

$$H(z) = \frac{z}{z - 0.9}$$

10. Perform the equalization of a channel with the following impulse response

$$h(k) = ku(k) - (2k - 9)u(k - 5) + (k - 9)u(k - 10)$$

Using a known training signal that consists of a binary (−1,1) random signal, generated by applying a white noise to a hard limiter (the output is 1 for positive input samples and −1 for negative). An additional Gaussian white noise with variance 10^{-2} is present at the channel output.

   (a) Apply the LMS with an appropriate μ and find the impulse response of an equalizer with 100 coefficients.
   (b) Convolve one of the equalizer's impulse response after convergence with the channel impulse response and comment on the result.

11. Under the assumption that the elements of $\mathbf{x}(k)$ are jointly Gaussian, show that (3.24) is valid.
12. In a system identification problem the input signal is generated by an autoregressive process given by

$$x(k) = -1.2x(k - 1) - 0.81x(k - 2) + n_x(k)$$

where $n_x(k)$ is zero-mean Gaussian white noise with variance such that $\sigma_x^2 = 1$. The unknown system is described by

$$H(z) = 1 + 0.9z^{-1} + 0.1z^{-2} + 0.2z^{-3}$$

The adaptive filter is also a third-order FIR filter, and the additional noise is zero-mean Gaussian white noise with variance $\sigma_n^2 = 0.04$. Using the LMS algorithm:

(a) Choose an appropriate μ, run an ensemble of 20 experiments, and plot the average learning curve.
(b) Plot the curve obtained using (3.41), (3.45), and (3.46), and compare the results.
(c) Compare the measured and theoretical values for the misadjustment.
(d) Calculate the time constants τ_{wi} and τ_{ei}, and the expected number of iterations to achieve convergence.

13. In a nonstationary environment the optimal coefficient vector is described by

$$\mathbf{w}_o(k) = -\lambda_1 \mathbf{w}_o(k-1) - \lambda_2 \mathbf{w}_o(k-2) + \mathbf{n}_{\mathbf{w}}(k)$$

where $\mathbf{n}_{\mathbf{w}}(k)$ is a vector whose elements are zero-mean Gaussian white processes with variance $\sigma_{\mathbf{w}}^2$. Calculate the elements of the lag-error vector.

14. Repeat the previous problem for

$$\mathbf{w}_o(k) = \lambda_w \mathbf{w}_o(k-1) + (1-\lambda_w)\mathbf{n}_{\mathbf{w}}(k)$$

15. The LMS algorithm is applied to identify a 7th-order time-varying unknown system whose coefficients are first-order Markov processes with $\lambda_{\mathbf{w}} = 0.999$ and $\sigma_{\mathbf{w}}^2 = 0.001$. The initial time-varying-system multiplier coefficients are

$$\mathbf{w}_o^T = [0.03490 \quad -0.011 \quad -0.06864 \quad 0.22391 \quad 0.55686 \quad 0.35798 \\ -0.0239 \quad -0.07594]$$

The input signal is Gaussian white noise with variance $\sigma_x^2 = 0.7$, and the measurement noise is also Gaussian white noise independent of the input signal and of the elements of $\mathbf{n}_{\mathbf{w}}(k)$, with variance $\sigma_n^2 = 0.01$.

(a) For $\mu = 0.05$, compute the excess MSE.
(b) Repeat (a) for $\mu = 0.01$.
(c) Compute μ_{opt} and comment if it can be used.

16. Simulate the experiment described in Problem 15, measure the excess MSE, and compare to the calculated results.
17. Reduce the value of $\lambda_{\mathbf{w}}$ to 0.97 in Problem 15, simulate, and comment on the results.

18. Suppose a 15th-order FIR digital filter with multiplier coefficients given below is identified through an adaptive FIR filter of the same order using the LMS algorithm.

    (a) Considering that fixed-point arithmetic is used, compute the expected value for $||\Delta \mathbf{w}(k)_Q||^2$ and $\xi(k)_Q$, and the probable number of iterations before the algorithm stops updating, for the following case:

Additional noise: white noise with variance	$\sigma_n^2 = 0.0015$
Coefficient wordlength:	$b_c = 16$ bits
Signal wordlength:	$b_d = 16$ bits
Input signal: Gaussian white noise with variance	$\sigma_x^2 = 0.7$
	$\mu = 0.01$

    Hint: Utilize the formulas for the time constant in the LMS algorithm and (15.28).

    (b) Simulate the experiment and plot the learning curves for the finite- and infinite-precision implementations.

    (c) Compare the simulated results with those obtained through the closed form formulas.

$$\begin{aligned}\mathbf{w}_o^T = [&0.0219360 \quad 0.0015786 \quad -0.0602449 \quad -0.0118907 \quad 0.1375379\\ &0.0574545 \quad -0.3216703 \quad -0.5287203 \quad -0.2957797 \quad 0.0002043\\ &0.290670 \quad -0.0353349 \quad -0.068210 \quad 0.0026067 \quad 0.0010333 \quad -0.0143593]\end{aligned}$$

19. Repeat the above problem for the following cases

    (a) $\sigma_n^2 = 0.01$, $b_c = 12$ bits, $b_d = 12$ bits, $\sigma_x^2 = 0.7$, $\mu = 2.0\ \ 10^{-3}$.
    (b) $\sigma_n^2 = 0.1$, $b_c = 10$ bits, $b_d = 10$ bits, $\sigma_x^2 = 0.8$, $\mu = 1.0\ \ 10^{-4}$.
    (c) $\sigma_n^2 = 0.05$, $b_c = 14$ bits, $b_d = 14$ bits, $\sigma_x^2 = 0.8$, $\mu = 2.0\ \ 10^{-3}$.

20. Find the optimal value of μ (μ_{opt}) that minimizes the excess MSE given in (15.32), and compute for $\mu = \mu_{\text{opt}}$ the expected value of $||\Delta \mathbf{w}(k)_Q||^2$ and $\xi(k)_Q$ for the examples described in Problem 19.
21. Repeat Problem 18 for the case where the input signal is a first-order Markov process with $\lambda_{\mathbf{x}} = 0.95$.
22. A digital channel model can be represented by the following impulse response:

$$\begin{aligned}[&-0.001 \quad -0.002 \quad 0.002 \quad 0.2 \quad 0.6 \quad 0.76 \quad 0.9 \quad 0.78 \quad 0.67 \quad 0.58\\ &0.45 \quad 0.3 \quad 0.2 \quad 0.12 \quad 0.06 \quad 0 \quad -0.2 \quad -1 \quad -2 \quad -1 \quad 0 \quad 0.1]\end{aligned}$$

The channel is corrupted by Gaussian noise with power spectrum given by

$$|S(\mathrm{e}^{\jmath\omega})|^2 = \kappa'|\omega|^{3/2}$$

where $\kappa' = 10^{-1.5}$. The training signal consists of independent binary samples (−1,1).

Design an FIR equalizer for this problem and use the LMS algorithm. Use a filter of order 50 and plot the learning curve.

23. For the previous problem, using the maximum of 51 adaptive filter coefficients, implement a DFE equalizer and compare the results with those obtained with the FIR filter. Again use the LMS algorithm.
24. Implement with fixed-point arithmetic the DFE equalizer of Problem 23, using the LMS algorithm with 12 bits of wordlength for data and coefficients.
25. Use the complex LMS algorithm to equalize a channel with the transfer function given below. The input signal is a four Quadrature Amplitude Modulation (QAM)[5] signal representing a randomly generated bit stream with the signal-to-noise ratio $\frac{\sigma_{\tilde{x}}^2}{\sigma_n^2} = 20$ at the receiver end, that is, $\tilde{x}(k)$ is the received signal without taking into consideration the additional channel noise. The adaptive filter has ten coefficients.

$$H(z) = (0.34 - 0.27\jmath) + (0.87 + 0.43\jmath)z^{-1} + (0.34 - 0.21\jmath)z^{-2}$$

(a) Calculate the upper bound for μ ($\mu_{\max}$) to guarantee the algorithm stability.
(b) Run the algorithm for $\mu_{\max}/2$, $\mu_{\max}/10$, and $\mu_{\max}/50$. Comment on the convergence behavior in each case.
(c) Plot the real versus imaginary parts of the received signal before and after equalization.
(d) Increase the number of coefficients to 20 and repeat the experiment in (c).

26. In a system identification problem the input signal is generated from a four QAM of the form

$$x(k) = x_{\text{re}}(k) + \jmath x_{\text{im}}(k)$$

where $x_{\text{re}}(k)$ and $x_{\text{im}}(k)$ assume values ±1 randomly generated. The unknown system is described by

$$H(z) = 0.32 + 0.21\jmath + (-0.3 + 0.7\jmath)z^{-1} + (0.5 - 0.8\jmath)z^{-2} + (0.2 + 0.5\jmath)z^{-3}$$

The adaptive filter is also a third-order complex FIR filter, and the additional noise is zero-mean Gaussian white noise with variance $\sigma_n^2 = 0.4$. Using the complex LMS algorithm, choose an appropriate μ, run an ensemble of 20 experiments, and plot the average learning curve.

[5] The M-ary QAM constellation points are represented in by $s_i = \tilde{a}_i + \jmath \tilde{b}_i$, with $\tilde{a}_i = \pm\tilde{d}, \pm 3\tilde{d}, \ldots, \pm(\sqrt{M} - 1)\tilde{d}$, and $\tilde{b}_i = \pm\tilde{d}, \pm 3\tilde{d}, \ldots, \pm(\sqrt{M} - 1)\tilde{d}$. The parameter $\tilde{d}$ is represents half of the distance between two points in the constellation.

References

1. B. Widrow, M.E. Hoff, Adaptive switching circuits. WESCOM Conv. Rec. **4**, 96–140 (1960)
2. B. Widrow, J.M. McCool, M.G. Larimore, C.R. Johnson Jr., Stationary and nonstationary learning characteristics of the LMS adaptive filters. Proc. IEEE **64**, 1151–1162 (1976)
3. G. Ungerboeck, Theory on the speed of convergence in adaptive equalizers for digital communication. IBM J. Res. Dev. **16**, 546–555 (1972)
4. J.E. Mazo, On the independence theory of equalizer convergence. Bell Syst. Tech. J. **58**, 963–993 (1979)
5. B. Widrow, S.D. Stearns, *Adaptive Signal Processing* (Prentice Hall, Englewood Cliffs, 1985)
6. S. Haykin, *Adaptive Filter Theory*, 4th edn. (Prentice Hall, Englewood Cliffs, 2002)
7. M.G. Bellanger, *Adaptive Digital Filters and Signal Analysis*, 2nd edn. (Marcel Dekker, Inc., New York, 2001)
8. D.C. Farden, Racking properties of adaptive signal processing algorithms. IEEE Trans. Acoust. Speech Signal Proc. **ASSP-29**, 439–446 (1981)
9. B. Widrow, E. Walach, On the statistical efficiency of the LMS algorithm with nonstationary inputs. IEEE Trans. Inform. Theor. **IT-30**, 211–221 (1984)
10. O. Macchi, Optimization of adaptive identification for time varying filters. IEEE Trans. Automat. Contr. **AC-31**, 283–287 (1986)
11. A. Benveniste, Design of adaptive algorithms for the tracking of time varying systems. Int. J. Adapt. Contr. Signal Process. **1**, 3–29 (1987)
12. W.A. Gardner, Nonstationary learning characteristics of the LMS algorithm. IEEE Trans. Circ. Syst. **CAS-34**, 1199–1207 (1987)
13. A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd edn. (McGraw Hill, New York, 1991)
14. F.J. Gantmacher, *The Theory of Matrices*, vol. 2 (Chelsea Publishing Company, New York, 1964)
15. G.H. Golub, C.F. Van Loan, *Matrix Computations*, 3rd edn. (John Hopkins University Press, Baltimore, 1996)
16. V. Solo, The limiting behavior of LMS. IEEE Trans. Acoust. Speech Signal Process. **37**, 1909–1922 (1989)
17. N.J. Bershad, O.M. Macchi, Adaptive recovery of a chirped sinusoid in noise, Part 2: Performance of the LMS algorithm. IEEE Trans. Signal Process. **39**, 595–602 (1991)
18. D.H. Brandwood, A complex gradient operator and its application in adaptive array theory. IEE Proc. Parts F and G **130**, 11–16 (1983)
19. A. Hjørungnes, D. Gesbert, Complex-valued matrix differentiation: Techniques and key results. IEEE Trans. Signal Process. **55**, 2740–2746 (2007)
20. D.G. Manolakis, V.K. Ingle, S.M. Kogon, *Statistical and Adaptive Signal Processing* (McGraw Hill, New York, 2000)
21. O. Macchi, E. Eweda, Second-order convergence analysis of stochastic adaptive linear filter. IEEE Trans. Automat. Contr. **AC-28**, 76–85 (1983)
22. V.H. Nascimento, A.H. Sayed, On the learning mechanism of adaptive filters. IEEE Trans. Signal Process. **48**, 1609–1625 (2000)
23. S. Florian, A. Feuer, Performance analysis of the LMS algorithm with a tapped delay line (two-dimensional case). IEEE Trans. Acoust. Speech Signal Process. **ASSP-34**, 1542–1549 (1986)
24. O.L. Frost III, An algorithm for linearly constrained adaptive array processing. Proc. IEEE **60**, 926–935 (1972)
25. J.A. Apolinário Jr., S. Werner, T.I. Laakso, P.S.R. Diniz, Constrained normalized adaptive filtering for CDMA mobile communications, in *Proceedings of 1998 EUSIPCO-European Signal Processing Conference*, Rhodes, Greece, Sept 1998, pp. 2053–2056
26. J.A. Apolinário Jr., M.L.R. de Campos, C.P. Bernal O, The constrained conjugate-gradient algorithm. IEEE Signal Process. Lett. **7**, 351–354 (2000)

27. M.L.R. de Campos, S. Werner, J.A. Apolinário Jr., Constrained adaptation algorithms employing Householder transformation. IEEE Trans. Signal Process. **50**, 2187–2195 (2002)
28. A. Feuer, E. Weinstein, Convergence analysis of LMS filters with uncorrelated Gaussian data. IEEE Trans. Acoust. Speech Signal Process. **ASSP-33**, 222–230 (1985)
29. D.T. Slock, On the convergence behavior of the LMS and normalized LMS algorithms. IEEE Trans. Signal Process. **40**, 2811–2825 (1993)
30. W.A. Sethares, D.A. Lawrence, C.R. Johnson Jr., R.R. Bitmead, Parameter drift in LMS adaptive filters. IEEE Trans. Acoust. Speech Signal Process. **ASSP-34**, 868–878 (1986)
31. S.C. Douglas, Exact expectation analysis of the LMS adaptive filter. IEEE Trans. Signal Process. **43**, 2863–2871 (1995)
32. H.J. Butterweck, Iterative analysis of the state-space weight fluctuations in LMS-type adaptive filters. IEEE Trans. Signal Process. **47**, 2558–2561 (1999)
33. B. Hassibi, A.H. Sayed, T. Kailath, H^{∞} optimality of the LMS Algorithm. IEEE Trans. Signal Process. **44**, 267–280 (1996)
34. O.J. Tobias, J.C.M. Bermudez, N.J. Bershad, Mean weight behavior of the filtered-X LMS algorithm. IEEE Trans. Signal Process. **48**, 1061–1075 (2000)
35. V. Solo, The error variance of LMS with time varying weights. IEEE Trans. Signal Process. **40**, 803–813 (1992)
36. S.U. Qureshi, Adaptive equalization. Proc. IEEE, **73**, 1349–1387 (1985)
37. M.L. Honig, Echo cancellation of voiceband data signals using recursive least squares and stochastic gradient algorithms. IEEE Trans. Comm. **COM-33**, 65–73 (1985)
38. V.H. Nascimento, J.C.M. Bermudez, Probability of divergence for the least-mean fourth algorithm. IEEE Trans. Signal Proces. **54**, 1376–1385 (2006)