5

CONVENTIONAL RLS ADAPTIVE FILTER

5.1 INTRODUCTION

Least-squares algorithms aim at the minimization of the sum of the squares of the difference between the desired signal and the model filter output [1]-[2]. When new samples of the incoming signals are received at every iteration, the solution for the least-squares problem can be computed in recursive form resulting in the recursive least-squares (RLS) algorithms. The conventional version of these algorithms will be the topic of the present chapter.

The RLS algorithms are known to pursue fast convergence even when the eigenvalue spread of the input signal correlation matrix is large. These algorithms have excellent performance when working in time-varying environments. All these advantages come with the cost of an increased computational complexity and some stability problems, which are not as critical in LMS-based algorithms [3]-[4].

Several properties related to the RLS algorithms are discussed including misadjustment, tracking behavior, which are verified through a number of simulation results.

Appendix C deals with the quantization effects in the conventional RLS algorithm. Appendix D provides an introduction to Kalman filters whose special case can be related to the RLS algorithms.

5.2 THE RECURSIVE LEAST-SQUARES ALGORITHM

The objective here is to choose the coefficients of the adaptive filter such that the output signal $y(k)$, during the period of observation, will match the desired signal as closely as possible in the least-squares sense. The minimization process requires the information of the input signal available so far. Also, the objective function we seek to minimize is deterministic.

The generic FIR adaptive filter realized in the direct form is shown in Fig. 5.1. The input signal information vector at a given instant k is given by

$$\mathbf{x}(k) = [x(k)\, x(k-1) \ldots x(k-N)]^T \tag{5.1}$$

P.S.R. Diniz, *Adaptive Filtering*, DOI: 10.1007/978-0-387-68606-6_5,

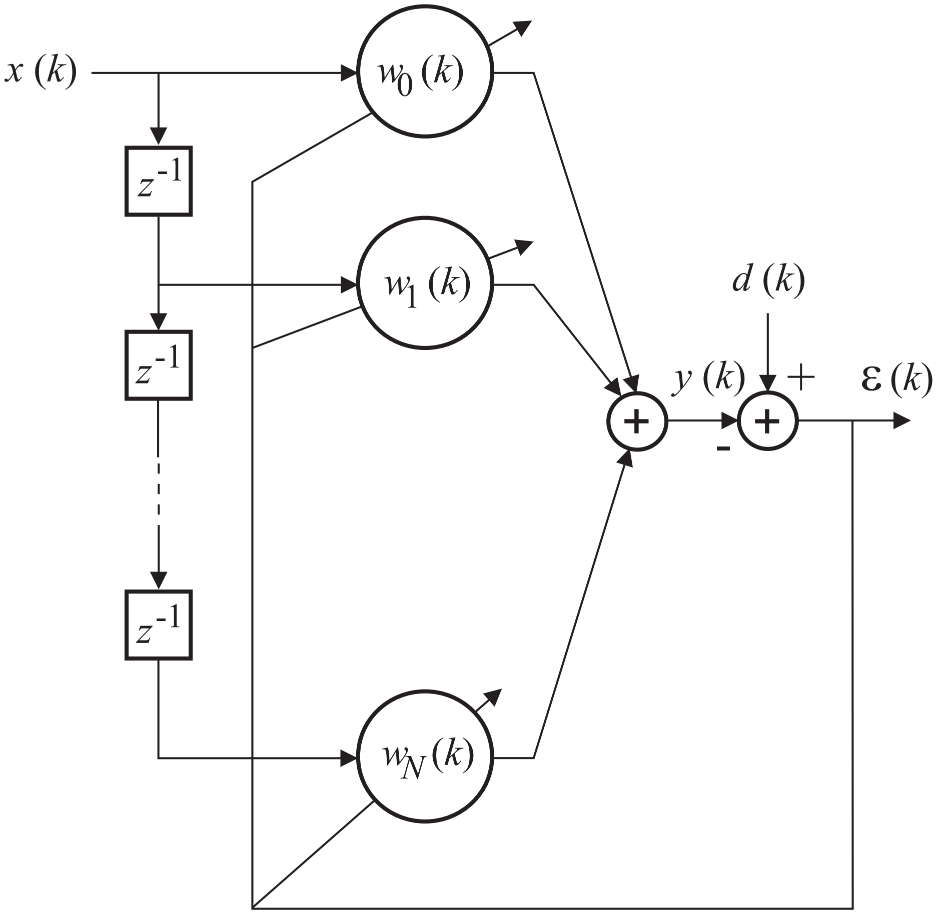


Figure 5.1 Adaptive FIR filter.

where N is the order of the filter. The coefficients $w_j(k)$, for $j = 0, 1, \ldots, N$, are adapted aiming at the minimization of a given objective function. In the case of least-squares algorithms, the objective function is deterministic and is given by

$$\begin{aligned}\xi^d(k) &= \sum_{i=0}^{k} \lambda^{k-i} \varepsilon^2(i) \\ &= \sum_{i=0}^{k} \lambda^{k-i} \left[d(i) - \mathbf{x}^T(i)\mathbf{w}(k)\right]^2 \end{aligned} \tag{5.2}$$

where $\mathbf{w}(k) = [w_o(k)\, w_1(k) \ldots w_N(k)]^T$ is the adaptive-filter coefficient vector and $\varepsilon(i)$ is the *a posteriori* output error[1] at instant i. The parameter λ is an exponential weighting factor that should be chosen in the range $0 \ll \lambda \leq 1$. This parameter is also called forgetting factor since the information of the distant past has an increasingly negligible effect on the coefficient updating.

It should be noticed that in the development of the LMS and LMS-based algorithms we utilized the *a priori* error. In the RLS algorithms $\varepsilon(k)$ is used to denote the *a posteriori* error whereas $e(k)$ denotes the *a priori* error. The *a posteriori* error will be our first choice in the development of the RLS-based algorithms.

As can be noted, each error consists of the difference between the desired signal and the filter output, using the most recent coefficients $\mathbf{w}(k)$. By differentiating $\xi^d(k)$ with respect to $\mathbf{w}(k)$, it follows that

$$\frac{\partial \xi^d(k)}{\partial \mathbf{w}(k)} = -2 \sum_{i=0}^{k} \lambda^{k-i} \mathbf{x}(i)[d(i) - \mathbf{x}^T(i)\mathbf{w}(k)] \tag{5.3}$$

By equating the result to zero, it is possible to find the optimal vector $\mathbf{w}(k)$ that minimizes the least-squares error, through the following relation:

$$-\sum_{i=0}^{k} \lambda^{k-i} \mathbf{x}(i)\mathbf{x}^T(i)\mathbf{w}(k) + \sum_{i=0}^{k} \lambda^{k-i} \mathbf{x}(i)d(i) = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

The resulting expression for the optimal coefficient vector $\mathbf{w}(k)$ is given by

$$\begin{aligned} \mathbf{w}(k) &= \left[\sum_{i=0}^{k} \lambda^{k-i} \mathbf{x}(i)\mathbf{x}^T(i)\right]^{-1} \sum_{i=0}^{k} \lambda^{k-i} \mathbf{x}(i)d(i) \\ &= \mathbf{R}_D^{-1}(k)\mathbf{p}_D(k) \end{aligned} \tag{5.4}$$

where $\mathbf{R}_D(k)$ and $\mathbf{p}_D(k)$ are called the deterministic correlation matrix of the input signal and the deterministic cross-correlation vector between the input and desired signals, respectively.

In equation (5.4) it was assumed that $\mathbf{R}_D(k)$ is nonsingular. However, if $\mathbf{R}_D(k)$ is singular a generalized inverse [1] should be used instead in order to obtain a solution for $\mathbf{w}(k)$ that minimizes $\xi^d(k)$. Since we are assuming that in most practical applications the input signal has persistence of excitation, the cases requiring generalized inverse are not discussed here. It should be mentioned that if the input signal is considered to be zero for $k < 0$ then $\mathbf{R}_D(k)$ will always be singular for $k < N$, i.e., during the initialization period. During this period, the optimal value of the coefficients can be calculated for example by the backsubstitution algorithm to be presented in subsection 9.2.1.

The straightforward computation of the inverse of $\mathbf{R}_D(k)$ results in an algorithm with computational complexity of $O[N^3]$. In the conventional RLS algorithm the computation of the inverse matrix is

[1] The *a posteriori* error is computed after the coefficient vector is updated, and taking into consideration the most recent input data vector $\mathbf{x}(k)$.

avoided through the use of the matrix inversion lemma [1], first presented in the previous chapter for the LMS-Newton algorithm. Using the matrix inversion lemma, see equation (4.51), the inverse of the deterministic correlation matrix can then be calculated in the following form

$$\mathbf{S}_D(k) = \mathbf{R}_D^{-1}(k) = \frac{1}{\lambda}\left[\mathbf{S}_D(k-1) - \frac{\mathbf{S}_D(k-1)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{S}_D(k-1)}{\lambda + \mathbf{x}^T(k)\mathbf{S}_D(k-1)\mathbf{x}(k)}\right] \tag{5.5}$$

The complete conventional RLS algorithm is described in Algorithm 5.1.

Algorithm 5.1

Conventional RLS Algorithm

```
Initialization
  $\mathbf{S}_D(-1) = \delta\mathbf{I}$
where $\delta$ can be the inverse of the input signal power estimate
  $\mathbf{p}_D(-1) = \mathbf{x}(-1) = [0\,0\ldots0]^T$
Do for $k \geq 0$ :
  $\mathbf{S}_D(k) = \frac{1}{\lambda}[\mathbf{S}_D(k-1) - \frac{\mathbf{S}_D(k-1)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{S}_D(k-1)}{\lambda+\mathbf{x}^T(k)\mathbf{S}_D(k-1)\mathbf{x}(k)}]$
  $\mathbf{p}_D(k) = \lambda\mathbf{p}_D(k-1) + d(k)\mathbf{x}(k)$
  $\mathbf{w}(k) = \mathbf{S}_D(k)\mathbf{p}_D(k)$
If necessary compute
  $y(k) = \mathbf{w}^T(k)\mathbf{x}(k)$
  $\varepsilon(k) = d(k) - y(k)$
```

An alternative way to describe the conventional RLS algorithm can be obtained if equation (5.4) is rewritten in the following form

$$\left[\sum_{i=0}^{k}\lambda^{k-i}\mathbf{x}(i)\mathbf{x}^T(i)\right]\mathbf{w}(k) = \lambda\left[\sum_{i=0}^{k-1}\lambda^{k-i-1}\mathbf{x}(i)d(i)\right] + \mathbf{x}(k)d(k) \tag{5.6}$$

By considering that $\mathbf{R}_D(k-1)\mathbf{w}(k-1) = \mathbf{p}_D(k-1)$, it follows that

$$\begin{aligned}\left[\sum_{i=0}^{k}\lambda^{k-i}\mathbf{x}(i)\mathbf{x}^T(i)\right]\mathbf{w}(k) &= \lambda\mathbf{p}_D(k-1) + \mathbf{x}(k)d(k)\\ &= \lambda\mathbf{R}_D(k-1)\mathbf{w}(k-1) + \mathbf{x}(k)d(k)\\ &= \left[\sum_{i=0}^{k}\lambda^{k-i}\mathbf{x}(i)\mathbf{x}^T(i) - \mathbf{x}(k)\mathbf{x}^T(k)\right]\mathbf{w}(k-1) + \mathbf{x}(k)d(k)\end{aligned} \tag{5.7}$$

where in the last equality the matrix $\mathbf{x}(k)\mathbf{x}^T(k)$ was added and subtracted inside square bracket on the right side of equation (5.7). Now, define the *a priori* error as

$$e(k) = d(k) - \mathbf{x}^T(k)\mathbf{w}(k-1) \tag{5.8}$$

By expressing $d(k)$ as a function of the *a priori* error and replacing the result in equation (5.7), after few manipulations, it can be shown that

$$\mathbf{w}(k) = \mathbf{w}(k-1) + e(k)\mathbf{S}_D(k)\mathbf{x}(k) \tag{5.9}$$

With equation (5.9), it is straightforward to generate an alternative conventional RLS algorithm as shown in Algorithm 5.2.

Algorithm 5.2

Alternative RLS Algorithm

```
Initialization
  $\mathbf{S}_D(-1) = \delta\mathbf{I}$
where $\delta$ can be the inverse of an estimate of the input signal power
  $\mathbf{x}(-1) = \mathbf{w}(-1) = [0\,0 \ldots 0]^T$
Do for $k \geq 0$
   $e(k) = d(k) - \mathbf{x}^T(k)\mathbf{w}(k-1)$
   $\boldsymbol{\psi}(k) = \mathbf{S}_D(k-1)\mathbf{x}(k)$
   $\mathbf{S}_D(k) = \frac{1}{\lambda}[\mathbf{S}_D(k-1) - \frac{\boldsymbol{\psi}(k)\boldsymbol{\psi}^T(k)}{\lambda+\boldsymbol{\psi}^T(k)\mathbf{x}(k)}]$
   $\mathbf{w}(k) = \mathbf{w}(k-1) + e(k)\mathbf{S}_D(k)\mathbf{x}(k)$
If necessary compute
   $y(k) = \mathbf{w}^T(k)\mathbf{x}(k)$
   $\varepsilon(k) = d(k) - y(k)$
```

In Algorithm 5.2, $\boldsymbol{\psi}(k)$ is an auxiliary vector required to reduce the computational burden defined by

$$\boldsymbol{\psi}(k) = \mathbf{S}_D(k-1)\mathbf{x}(k) \tag{5.10}$$

Further reduction in the number of divisions is possible if an additional auxiliary vector is used, defined as

$$\boldsymbol{\phi}(k) = \frac{\boldsymbol{\psi}(k)}{\lambda + \boldsymbol{\psi}^T(k)\mathbf{x}(k)} \tag{5.11}$$

This vector can be used to update $\mathbf{S}_D(k)$ as follows:

$$\mathbf{S}_D(k) = \frac{1}{\lambda}\left[\mathbf{S}_D(k-1) - \boldsymbol{\psi}(k)\boldsymbol{\phi}^T(k)\right] \tag{5.12}$$

As will be discussed, the above relation can lead to stability problems in the RLS algorithm.

5.3 PROPERTIES OF THE LEAST-SQUARES SOLUTION

In this section, some properties related to the least-squares solution are discussed in order to give some insight to the algorithm behavior in several situations to be discussed later on.

5.3.1 Orthogonality Principle

Define the matrices $\mathbf{X}(k)$ and $\mathbf{d}(k)$ that contain all the information about the input signal vector $\mathbf{x}(k)$ and the desired signal vector $d(k)$ as follows:

$$\begin{aligned}\mathbf{X}(k) &= \begin{bmatrix} x(k) & \lambda^{1/2}x(k-1) & \cdots & \lambda^{(k-1)/2}x(1) & \lambda^{k/2}x(0) \\ x(k-1) & \lambda^{1/2}x(k-2) & \cdots & \lambda^{(k-1)/2}x(0) & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ x(k-N) & \lambda^{1/2}x(k-N-1) & \cdots & 0 & 0 \end{bmatrix} \\ &= [\mathbf{x}(k)\ \lambda^{1/2}\mathbf{x}(k-1) \ldots \lambda^{k/2}\mathbf{x}(0)] \qquad (5.13) \\ \mathbf{d}(k) &= [d(k)\ \lambda^{1/2}d(k-1) \ldots \lambda^{k/2}d(0)]^T \qquad (5.14)\end{aligned}$$

where $\mathbf{X}(k)$ is $(N+1) \times (k+1)$ and $\mathbf{d}(k)$ is $(k+1) \times 1$.

By using the matrices above defined it is possible to replace the least-squares solution of equation (5.4) by the following relation

$$\mathbf{X}(k)\mathbf{X}^T(k)\mathbf{w}(k) = \mathbf{X}(k)\mathbf{d}(k) \qquad (5.15)$$

The product $\mathbf{X}^T(k)\mathbf{w}(k)$ forms a vector including all the adaptive-filter outputs when the coefficients are given by $\mathbf{w}(k)$. This vector corresponds to an estimate of $\mathbf{d}(k)$. Hence, defining

$$\mathbf{y}(k) = \mathbf{X}^T(k)\mathbf{w}(k) = [y(k)\ \lambda^{1/2}y(k-1) \ldots \lambda^{k/2}y(0)]^T \qquad (5.16)$$

it follows from equation (5.15) that

$$\mathbf{X}(k)\mathbf{X}^T(k)\mathbf{w}(k) - \mathbf{X}(k)\mathbf{d}(k) = \mathbf{X}(k)[\mathbf{y}(k) - \mathbf{d}(k)] = \mathbf{0} \qquad (5.17)$$

This relation means that the weighted-error vector given by

$$\boldsymbol{\varepsilon}(k) = \begin{bmatrix} \varepsilon(k) \\ \lambda^{1/2}\varepsilon(k-1) \\ \vdots \\ \lambda^{k/2}\varepsilon(0) \end{bmatrix} = \mathbf{d}(k) - \mathbf{y}(k) \qquad (5.18)$$

is in the null space of $\mathbf{X}(k)$, i.e., the weighted-error vector is orthogonal to all row vectors of $\mathbf{X}(k)$. This justifies the fact that (5.15) is often called normal equation. A geometrical interpretation can easily be given for a least-squares problem solution with a single coefficient filter.

Example 5.1

Suppose that $\lambda = 1$ and that the following signals are involved in the least-squares problem

$$\mathbf{d}(1) = \begin{bmatrix} 0.5 \\ 1.5 \end{bmatrix} \quad \mathbf{X}(1) = [1 \quad -2]$$

The optimal coefficient is given by

$$\begin{aligned}\mathbf{X}(1)\mathbf{X}^T(1)\mathbf{w}(1) &= [1 \quad -2]\begin{bmatrix} 1 \\ -2 \end{bmatrix} \quad \mathbf{w}(1) \\ &= \mathbf{X}(1)\mathbf{d}(1) \\ &= [1 \quad -2]\begin{bmatrix} 0.5 \\ 1.5 \end{bmatrix}\end{aligned}$$

After performing the calculations the result is

$$\mathbf{w}(1) = -\frac{1}{2}$$

The output of the adaptive filter with coefficient given by $\mathbf{w}(1)$ is

$$\mathbf{y}(1) = \begin{bmatrix} -\frac{1}{2} \\ 1 \end{bmatrix}$$

Note that

$$\begin{aligned}\mathbf{X}(1)[\mathbf{y}(1) - \mathbf{d}(1)] &= [1 \quad -2]\begin{bmatrix} -1 \\ -0.5 \end{bmatrix} \\ &= 0\end{aligned}$$

Fig. 5.2 illustrates the fact that $\mathbf{y}(1)$ is the projection of $\mathbf{d}(1)$ in the $\mathbf{X}(1)$ direction. In the general case we can say that the vector $\mathbf{y}(k)$ is the projection of $\mathbf{d}(k)$ onto the subspace spanned by the rows of $\mathbf{X}(k)$.

□

5.3.2 Relation Between Least-Squares and Wiener Solutions

When $\lambda = 1$ the matrix $\frac{1}{k+1}\mathbf{R}_D(k)$ for large k is a consistent estimate of the input signal autocorrelation matrix $\mathbf{R}$, if the process from which the input signal was taken is ergodic. The same observation is valid for the vector $\frac{1}{k+1}\mathbf{p}_D(k)$ related to $\mathbf{p}$ if the desired signal is also ergodic. In this case,

$$\mathbf{R} = \lim_{k\to\infty} \frac{1}{k+1} \sum_{i=0}^{k} \mathbf{x}(i)\mathbf{x}^T(i) = \lim_{k\to\infty} \frac{1}{k+1}\mathbf{R}_D(k) \tag{5.19}$$

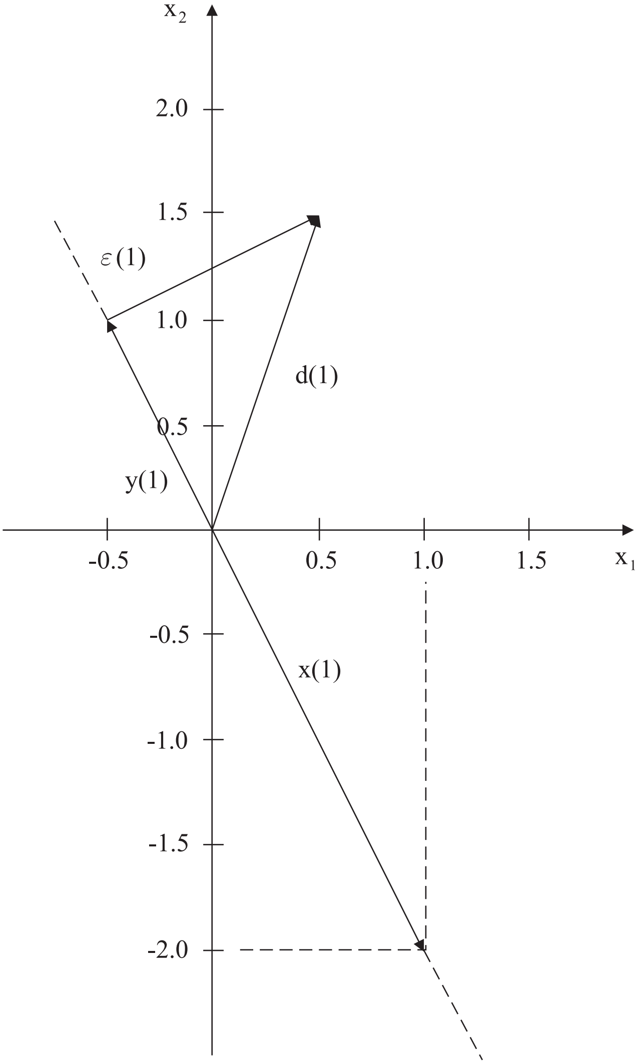


Figure 5.2 Geometric interpretation of least-squares solution.

and

$$\mathbf{p} = \lim_{k\to\infty} \frac{1}{k+1} \sum_{i=0}^{k} \mathbf{x}(i)d(i) = \lim_{k\to\infty} \frac{1}{k+1} \mathbf{p}_D(k) \tag{5.20}$$

It can then be shown that

$$\mathbf{w}(k) = \mathbf{R}_D^{-1}(k)\mathbf{p}_D(k) = \mathbf{R}^{-1}\mathbf{p} = \mathbf{w}_o \tag{5.21}$$

when k tends to infinity. This result indicates that the least-squares solution tends to the Wiener solution if the signals involved are ergodic and stationary. The stationarity requirement is due to the fact that the estimate of **R** given by equation (5.19) is not sensitive to any changes in **R** for large values of k. If the input signal is nonstationary $\mathbf{R}_D(k)$ is a biased estimate for **R**. Note that in this case **R** is time varying.

5.3.3 Influence of the Deterministic Autocorrelation Initialization

The initialization of $\mathbf{S}_D(-1) = \delta\mathbf{I}$ causes a bias in the coefficients estimated by the adaptive filter. Suppose that the initial value given to $\mathbf{R}_D(k)$ is taken into account in the actual RLS solution as follows:

$$\begin{aligned}\sum_{i=-1}^{k} \lambda^{k-i}\mathbf{x}(i)\mathbf{x}^T(i)\mathbf{w}(k) &= [\sum_{i=0}^{k} \lambda^{k-i}\mathbf{x}(i)\mathbf{x}^T(i) + \frac{\lambda^{k+1}}{\delta}\mathbf{I}]\mathbf{w}(k) \\ &= \mathbf{p}_D(k) \qquad (5.22)\end{aligned}$$

By recognizing that the deterministic autocorrelation matrix leading to an unbiased solution does not include the initialization matrix, we now examine the influence of this matrix. By multiplying $\mathbf{S}_D(k) = \mathbf{R}_D^{-1}(k)$ on both sides of equation (5.22), and by considering $k \rightarrow \infty$, it can be concluded that

$$\mathbf{w}(k) + \frac{\lambda^{k+1}}{\delta}\mathbf{S}_D(k)\mathbf{w}(k) = \mathbf{w}_o \qquad (5.23)$$

where $\mathbf{w}_o$ is the optimal solution for the RLS algorithm.

The bias caused by the initialization of $\mathbf{S}_D(k)$ is approximately

$$\mathbf{w}(k) - \mathbf{w}_o \approx -\frac{\lambda^{k+1}}{\delta}\mathbf{S}_D(k)\mathbf{w}_o \qquad (5.24)$$

For $\lambda < 1$, it is straightforward to conclude that the bias tends to zero as k tends to infinity. On the other hand, when $\lambda = 1$ the elements of $\mathbf{S}_D(k)$ get smaller when the number of iterations increase, as a consequence this matrix approaches a null matrix for large k.

The RLS algorithm would reach the optimum solution for the coefficients after $N+1$ iterations if no measurement noise is present, and the influence of the initialization matrix $\mathbf{S}_D(-1)$ is negligible at this point. This result follows from the fact that after $N+1$ iterations, the input signal vector has enough information to allow the adaptive algorithm to identify the coefficients of the unknown system. In other words, enough information means the tap delay line is filled with information of the input signal.

5.3.4 Steady-State Behavior of the Coefficient Vector

In order to understand better the steady-state behavior of the adaptive-filter coefficients, suppose that an FIR filter with coefficients given by $\mathbf{w}_o$ is being identified by an adaptive FIR filter of the same order employing an LS algorithm. Also assume that a measurement noise signal $n(k)$ is added to the desired signal before the error signal is calculated as follows:

$$d(k) = \mathbf{w}_o^T\mathbf{x}(k) + n(k) \qquad (5.25)$$

where the additional noise is considered to be a white noise with zero mean and variance given by σ_n^2.

Given the adaptive-filter input vectors $\mathbf{x}(k)$, for $k = 0, 1, \ldots$, we are interested in calculating the average values of the adaptive-filter coefficients $w_i(k)$, for $i = 0, 1, \ldots, N$. The desired result is the following equality valid for $k \geq N$.

$$\begin{aligned} E[\mathbf{w}(k)] &= E\left\{\left[\mathbf{X}(k)\mathbf{X}^T(k)\right]^{-1}\mathbf{X}(k)\mathbf{d}(k)\right\} \\ &= E\left\{\left[\mathbf{X}(k)\mathbf{X}^T(k)\right]^{-1}\mathbf{X}(k)[\mathbf{X}^T(k)\mathbf{w}_o + \mathbf{n}(k)]\right\} \\ &= E\left\{\left[\mathbf{X}(k)\mathbf{X}^T(k)\right]^{-1}\mathbf{X}(k)\mathbf{X}^T(k)\mathbf{w}_o\right\} = \mathbf{w}_o \end{aligned} \tag{5.26}$$

where $\mathbf{n}(k) = [n(k)\ \lambda^{1/2}n(k-1)\ \lambda n(k-2)\ \ldots\ \lambda^{k/2}n(0)]^T$ is the noise vector, whose elements were considered orthogonal to the input signal. The above equation shows that the estimate given by the LS algorithm is an unbiased estimate when $\lambda \leq 1$.

A more accurate analysis reveals the behavior of the adaptive-filter coefficients during the transient period. The error in the filter coefficients can be described by the following $(N+1) \times 1$ vector

$$\Delta\mathbf{w}(k) = \mathbf{w}(k) - \mathbf{w}_o \tag{5.27}$$

It follows from equation (5.7) that

$$\mathbf{R}_D(k)\mathbf{w}(k) = \lambda\mathbf{R}_D(k-1)\mathbf{w}(k-1) + \mathbf{x}(k)d(k) \tag{5.28}$$

Defining the minimum output error as

$$e_o(k) = d(k) - \mathbf{x}^T(k)\mathbf{w}_o \tag{5.29}$$

and replacing $d(k)$ in equation (5.28), it can be deduced that

$$\mathbf{R}_D(k)\Delta\mathbf{w}(k) = \lambda\mathbf{R}_D(k-1)\Delta\mathbf{w}(k-1) + \mathbf{x}(k)e_o(k) \tag{5.30}$$

where the following relation was used

$$\mathbf{R}_D(k) = \lambda\mathbf{R}_D(k-1) + \mathbf{x}(k)\mathbf{x}^T(k) \tag{5.31}$$

The solution of equation (5.30) is given by

$$\Delta\mathbf{w}(k) = \lambda^{k+1}\mathbf{S}_D(k)\mathbf{R}_D(-1)\Delta\mathbf{w}(-1) + \mathbf{S}_D(k)\sum_{i=0}^{k}\lambda^{k-i}\mathbf{x}(i)e_o(i) \tag{5.32}$$

By replacing $\mathbf{R}_D(-1)$ by $\frac{1}{\delta}\,\mathbf{I}$ and taking the expected value of the resulting equation, it follows that

$$E[\Delta\mathbf{w}(k)] = \frac{\lambda^{k+1}}{\delta}E[\mathbf{S}_D(k)]\Delta\mathbf{w}(-1) + E[\mathbf{S}_D(k)\sum_{i=0}^{k}\lambda^{k-i}\mathbf{x}(i)e_o(i)] \tag{5.33}$$

Since $\mathbf{S}_D(k)$ is dependent on all past input signal vectors, becoming relatively invariant when the number of iterations increase, the contribution of any individual $\mathbf{x}(i)$ can be considered negligible. Also, due to the orthogonality principle, $e_o(i)$ can also be considered uncorrelated to all elements of $\mathbf{x}(i)$. This means that the last vector in equation (5.33) cannot have large element values. On the other hand, the first vector in equation (5.33) can have large element values only during the initial convergence, since as $k \to \infty$, $\lambda^{k+1} \to 0$ and $\mathbf{S}_D(k)$ is expected to have a nonincreasing behavior, i.e., $\mathbf{R}_D(k)$ is assumed to remain positive definite as $k \to \infty$ and the input signal power does not become too small. The above discussion leads to the conclusion that the adaptive-filter coefficients tend to the optimal values in $\mathbf{w}_o$ almost independently from the eigenvalue spread of the input signal correlation matrix.

If we consider the spectral decomposition of the matrix $E[\mathbf{S}_D(k)]$ (see equation (2.65)), the dependency on the eigenvalues of $\mathbf{R}$ can be easily accounted for in the simple case of $\lambda = 1$. Applying the expected value operator to the relation of equation (5.19), we can infer that

$$E[\mathbf{S}_D(k)] \approx \frac{\mathbf{R}^{-1}}{(k+1)} \tag{5.34}$$

for large k. Now consider the slowest decaying mode of the spectral decomposition of $E[\mathbf{S}_D(k)]$ given by

$$\mathbf{S}_{D_{\max}} = \frac{\mathbf{q}_{\min}\mathbf{q}_{\min}^T}{(k+1)\lambda_{\min}} \tag{5.35}$$

where $\lambda_{\min}$ is the smallest eigenvalue of $\mathbf{R}$ and $\mathbf{q}_{\min}$ is the corresponding eigenvector. Applying this result to equation (5.33), with $\lambda = 1$, we can conclude that the value of the minimum eigenvalue affects the convergence of the filter coefficients only in the first few iterations, because the term $k+1$ in the denominator reduces the values of the elements of $\mathbf{S}_{D_{\max}}$.

Further interesting properties of the coefficients generated by the LS algorithm are:

- The estimated coefficients are the best linear unbiased solution to the identification problem [1], in the sense that no other unbiased solution generated by alternative approaches has lower variance.
- If the additive noise is normally distributed the LS solution reaches the Cramer-Rao lower bound, resulting in a minimum-variance unbiased solution [1]. The Cramer-Rao lower bound establishes a lower bound to the coefficient-error-vector covariance matrix for any unbiased estimator of the optimal parameter vector $\mathbf{w}_o$.

5.3.5 Coefficient-Error-Vector Covariance Matrix

So far, we have shown that the estimation parameters in the vector $\mathbf{w}(k)$ converge on average to their optimal value of the vector $\mathbf{w}_o$. However, it is essential to analyze the coefficient-error-vector covariance matrix in order to determine how good is the obtained solution, in the sense that we are measuring how far the parameters wander around the optimal solution.

Using the same convergence assumption of the last section, it will be shown here that for $\lambda = 1$ the coefficient-error-vector covariance matrix is given by

$$\text{cov}[\Delta\mathbf{w}(k)] = E\left[(\mathbf{w}(k) - \mathbf{w}_o)(\mathbf{w}(k) - \mathbf{w}_o)^T\right] = \sigma_n^2 E[\mathbf{S}_D(k)] \tag{5.36}$$

Proof:

First note that by using equations (5.4) and (5.15), the following relations are verified

$$\begin{aligned}
\mathbf{w}(k) - \mathbf{w}_o &= \mathbf{S}_D(k)\mathbf{p}_D(k) - \mathbf{S}_D(k)\mathbf{S}_D^{-1}(k)\mathbf{w}_o & (5.37)\\
&= \left[\mathbf{X}(k)\mathbf{X}^T(k)\right]^{-1}\mathbf{X}(k)\left[\mathbf{d}(k) - \mathbf{X}^T(k)\mathbf{w}_o\right] & (5.38)\\
&= \left[\mathbf{X}(k)\mathbf{X}^T(k)\right]^{-1}\mathbf{X}(k)\mathbf{n}(k) & (5.39)
\end{aligned}$$

where $\mathbf{n}(k) = [n(k)\ \lambda^{1/2}n(k-1)\ \lambda n(k-2)\ \ldots\ \lambda^{k/2}n(0)]^T$.

Applying the last equation to the covariance of the coefficient-error-vector it follows that

$$\begin{aligned}
\text{cov}[\Delta\mathbf{w}(k)] &= E\left\{\left[\mathbf{X}(k)\mathbf{X}^T(k)\right]^{-1}\mathbf{X}(k)E[\mathbf{n}(k)\mathbf{n}^T(k)]\mathbf{X}^T(k)\left[\mathbf{X}(k)\mathbf{X}^T(k)\right]^{-1}\right\}\\
&= E\left\{\sigma_n^2\mathbf{S}_D(k)\mathbf{X}(k)\Lambda\mathbf{X}^T(k)\mathbf{S}_D(k)\right\}
\end{aligned}$$

where

$$\Lambda = \begin{bmatrix} 1 & & & & \\ & \lambda & & \mathbf{0} & \\ & & \lambda^2 & & \\ & \mathbf{0} & & \ddots & \\ & & & & \lambda^k \end{bmatrix}$$

For $\lambda = 1$, $\Lambda = \mathbf{I}$, it follows that

$$\begin{aligned}
\text{cov}[\Delta\mathbf{w}(k)] &= E\left[\sigma_n^2\mathbf{S}_D(k)\mathbf{X}(k)\mathbf{X}^T(k)\mathbf{S}_D(k)\right]\\
&= E\left[\sigma_n^2\mathbf{S}_D(k)\mathbf{R}_D(k)\mathbf{S}_D(k)\right]\\
&= \sigma_n^2 E\left[\mathbf{S}_D(k)\right]
\end{aligned}$$

□

Therefore, when $\lambda = 1$, the coefficient-error-vector covariance matrix tends to decrease its norm as time progresses since $\mathbf{S}_D(k)$ is also norm decreasing. The variance of the additional noise $n(k)$ influences directly the norm of the covariance matrix.

5.3.6 Behavior of the Error Signal

It is important to understand how the error signal behaves in the RLS algorithm. When a measurement noise is present in the adaptive-filtering process, the *a priori* error signal is given by

$$e(k) = d'(k) - \mathbf{w}^T(k-1)\mathbf{x}(k) + n(k) \tag{5.40}$$

where $d'(k) = \mathbf{w}_o^T\mathbf{x}(k)$ is the desired signal without measurement noise.

Again if the input signal is considered known (conditional expectation), then

$$\begin{aligned} E[e(k)] &= E[d'(k)] - E[\mathbf{w}^T(k-1)]\mathbf{x}(k) + E[n(k)] \\ &= E[\mathbf{w}_o^T - \mathbf{w}_o^T]\mathbf{x}(k) + E[n(k)] \\ &= E[n(k)] \end{aligned} \tag{5.41}$$

assuming that the adaptive-filter order is sufficient to model perfectly the desired signal.

From equation (5.41), it can be concluded that if the noise signal has zero mean then

$$E[e(k)] = 0$$

It is also important to assess the minimum mean value of the squared error that is reachable using an RLS algorithm. The minimum mean-square error (MSE) in the presence of external uncorrelated noise is given by

$$\xi_{\min} = E[e^2(k)] = E[e_o^2(k)] = E[n^2(k)] = \sigma_n^2 \tag{5.42}$$

where it is assumed that the adaptive-filter multiplier coefficients were frozen at their optimum values and that the number of coefficients of the adaptive filter is sufficient to model the desired signal. In the conditions described the *a priori* error corresponds to the minimum output error as defined in equation (5.29). It should be noted, however, that if the additive noise is correlated with the input and the desired signals, a more complicated expression for the MSE results, accounting for the referred correlation.

When employing the *a posteriori* error the value of minimum MSE, denoted by $\xi_{\mathrm{min,p}}$, differs from the corresponding value related to the *a priori* error. First note that by using equation (5.39), the following relation is verified

$$\Delta\mathbf{w}(k) = \mathbf{S}_D(k)\mathbf{X}(k)\mathbf{n}(k) \tag{5.43}$$

When a measurement noise is present in the adaptive-filtering process, the *a posteriori* error signal is given by

$$\varepsilon(k) = d'(k) - \mathbf{w}^T(k)\mathbf{x}(k) + n(k) = -\Delta\mathbf{w}^T(k)\mathbf{x}(k) + e_o(k) \tag{5.44}$$

The expression for the MSE related to the *a posteriori* error is then given by

$$\begin{aligned} \xi(k) &= E[\varepsilon^2(k)] \\ &= E[e_o^2(k)] - 2E[\mathbf{x}^T(k)\Delta\mathbf{w}(k)e_o(k)] + E[\Delta\mathbf{w}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)\Delta\mathbf{w}(k)] \end{aligned} \tag{5.45}$$

By replacing the expression (5.43) in equation (5.45) above, the following relations follow

$$\begin{aligned}
\xi(k) =& E[e_o^2(k)] - 2E[\mathbf{x}^T(k)\mathbf{S}_D(k)\mathbf{X}(k)\mathbf{n}(k)e_o(k)] + E[\Delta\mathbf{w}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)\Delta\mathbf{w}(k)] \\
=& \sigma_n^2 - 2E[\mathbf{x}^T(k)\mathbf{S}_D(k)\mathbf{X}(k)] \begin{bmatrix} \sigma_n^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + E[\Delta\mathbf{w}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)\Delta\mathbf{w}(k)] \\
=& \sigma_n^2 - 2E[\mathbf{x}^T(k)\mathbf{S}_D(k)\mathbf{x}(k)]\sigma_n^2 + E[\Delta\mathbf{w}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)\Delta\mathbf{w}(k)] \\
=& \xi_{\text{min,p}} + E[\Delta\mathbf{w}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)\Delta\mathbf{w}(k)]
\end{aligned} \tag{5.46}$$

where in the second equality it was considered that the additional noise is uncorrelated with the input signal and that $e_o(k) = n(k)$. This equality occurs when the adaptive filter has sufficient order to identify the unknown system.

Note that $\xi_{\text{min,p}}$ related to the *a posteriori* error in equation (5.46) is not the same as minimum MSE of the *a priori* error, denoted in this book by ξ_{min}. The last term, that is $E[\Delta\mathbf{w}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)\Delta\mathbf{w}(k)]$, in equation (5.46) determines the excess MSE of the RLS algorithm.

It is possible to verify that the following expressions for $\xi_{\text{min,p}}$ are accurate approximations

$$\begin{aligned}
\xi_{\text{min,p}} &= \{1 - 2E[\mathbf{x}^T(k)\mathbf{S}_D(k)\mathbf{x}(k)]\}\sigma_n^2 \\
&= \left\{1 - 2\text{tr}\left[E\left(\mathbf{S}_D(k)\mathbf{x}(k)\mathbf{x}^T(k)\right)\right]\right\}\sigma_n^2 \\
&= \left\{1 - 2\text{tr}\left[\frac{1-\lambda}{1-\lambda^{k+1}}\mathbf{I}\right]\right\}\sigma_n^2 \\
&= \left\{1 - 2(N+1)\left[\frac{1-\lambda}{1-\lambda^{k+1}}\right]\right\}\sigma_n^2 \\
&= \left\{1 - 2(N+1)\left[\frac{1}{1+\lambda+\lambda^2+\cdots+\lambda^k}\right]\right\}\sigma_n^2
\end{aligned} \tag{5.47}$$

In the above expression, it is considered that $\mathbf{S}_D(k)$ is slowly varying as compared to $\mathbf{x}(k)$ when $\lambda \to 1$, such that

$$E[\mathbf{S}_D(k)\mathbf{x}(k)\mathbf{x}^T(k)] \approx E\left[\mathbf{S}_D(k)\right]E\left[\mathbf{x}(k)\mathbf{x}^T(k)\right]$$

and that by using equation (5.55)

$$E\left[\mathbf{S}_D(k)\mathbf{x}(k)\mathbf{x}^T(k)\right] \approx \frac{1-\lambda}{1-\lambda^{k+1}}\mathbf{I}$$

Equation (5.47) applies to the case where $\lambda < 1$, and as can be observed from the term multiplying $N+1$ there is a transient for small k which dies away when the number of iterations increases[2]. If we fit the decrease in the term multiplying $N+1$ at each iteration to an exponential envelop, the

[2] The expression for $\xi_{\text{min,p}}$ can be negative, however, $\xi(k)$ is always non negative.

time constant will be $\frac{1}{\lambda^{k+1}}$. Unlike the LMS algorithm, this time constant is time varying and is not related to the eigenvalue spread of the input signal correlation matrix.

Example 5.2

Repeat the equalization problem of example 3.1 using the RLS algorithm.

(a) Using $\lambda = 0.99$, run the algorithm and save matrix $\mathbf{S}_D(k)$ at iteration 500 and compare with the inverse of the input signal correlation matrix.

(b) Plot the convergence path for the RLS algorithm on the MSE surface.

Solution:

(a) The inverse of matrix **R**, as computed in the example 3.1, is given by

$$\mathbf{R}^{-1} = 0.45106 \begin{bmatrix} 1.6873 & 0.7937 \\ 0.7937 & 1.6873 \end{bmatrix} = \begin{bmatrix} 0.7611 & 0.3580 \\ 0.3580 & 0.7611 \end{bmatrix}$$

The initialization matrix $\mathbf{S}_D(-1)$ is a diagonal matrix with the diagonal elements equal to 0.1. The matrix $\mathbf{S}_D(k)$ at the 500th iteration, obtained by averaging the results of 30 experiments, is

$$\mathbf{S}_D(500) = \begin{bmatrix} 0.0078 & 0.0037 \\ 0.0037 & 0.0078 \end{bmatrix}$$

Also, the obtained values of the deterministic cross-correlation vector is

$$\mathbf{p}_D(500) \;=\; \begin{bmatrix} 95.05 \\ 46.21 \end{bmatrix}$$

Now, we divide each element of the matrix $\mathbf{R}^{-1}$ by

$$\frac{1-\lambda^{k+1}}{1-\lambda} = 99.34$$

since in a stationary environment $E[\mathbf{S}_D(k)] = \frac{1-\lambda}{1-\lambda^{k+1}}\mathbf{R}^{-1}$, see equation (5.55) for a formal proof.

The resulting matrix is

$$\frac{1}{99.34}\mathbf{R}^{-1} = \begin{bmatrix} 0.0077 & 0.0036 \\ 0.0036 & 0.0077 \end{bmatrix}$$

As can be noted the values of the elements of the above matrix are close to the average values of the corresponding elements of matrix $\mathbf{S}_D(500)$.

Similarly, if we multiply the cross-correlation vector **p** by 99.34, the result is

$$99.34\mathbf{p} = \begin{bmatrix} 94.61 \\ 47.31 \end{bmatrix}$$

The values of the elements of this vector are also close to the corresponding elements of $\mathbf{p}_D(500)$.

(b) The convergence path of the RLS algorithm on the MSE surface is depicted in Fig. 5.3. The reader should notice that the RLS algorithm approaches the minimum using large steps when the coefficients of the adaptive filter are far away from the optimum solution.

□

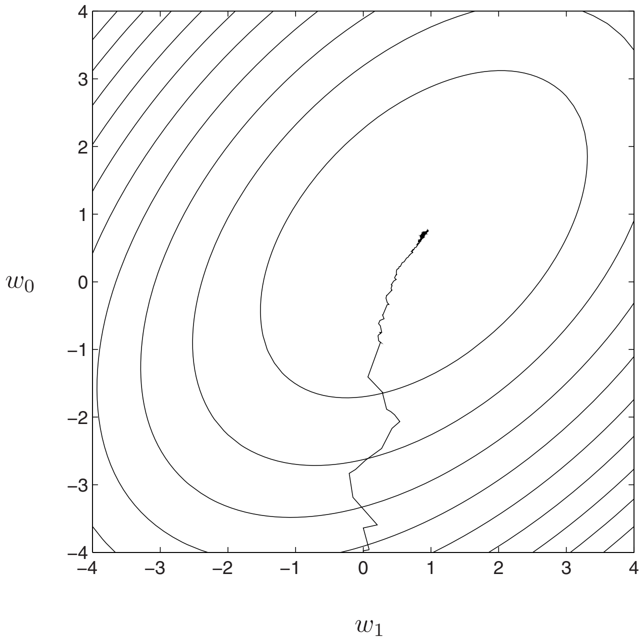


Figure 5.3 Convergence path of the RLS adaptive filter.

5.3.7 Excess Mean-Square Error and Misadjustment

In a practical implementation of the recursive least-squares algorithm, the best estimation for the unknown parameter vector is given by $\mathbf{w}(k)$, whose expected value is $\mathbf{w}_o$. However, there is always an excess MSE at the output caused by the error in the coefficient estimation, namely $\Delta\mathbf{w}(k) =$

$\mathbf{w}(k) - \mathbf{w}_o$. The mean-square error is (see equation (5.46))

$$\begin{aligned}\xi(k) &= \xi_{\mathrm{min,p}} + E\{[\mathbf{w}(k) - \mathbf{w}_o]^T \mathbf{x}(k)\mathbf{x}^T(k)[\mathbf{w}(k) - \mathbf{w}_o]\} \\ &= \xi_{\mathrm{min,p}} + E[\Delta\mathbf{w}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)\Delta\mathbf{w}(k)] \end{aligned} \tag{5.48}$$

Now considering that $\Delta w_j(k)$, for $j = 0, 1, \ldots, N$, are random variables with zero mean and independent of $\mathbf{x}(k)$, the MSE can be calculated as follows

$$\begin{aligned}\xi(k) &= \xi_{\mathrm{min,p}} + E[\Delta\mathbf{w}^T(k)\mathbf{R}\Delta\mathbf{w}(k)] \\ &= \xi_{\mathrm{min,p}} + E\{\mathrm{tr}[\mathbf{R}\Delta\mathbf{w}(k)\Delta\mathbf{w}^T(k)]\} \\ &= \xi_{\mathrm{min,p}} + \mathrm{tr}\{\mathbf{R}E[\Delta\mathbf{w}(k)\Delta\mathbf{w}^T(k)]\} \\ &= \xi_{\mathrm{min,p}} + \mathrm{tr}\{\mathbf{R}\mathrm{cov}[\Delta\mathbf{w}(k)]\} \end{aligned} \tag{5.49}$$

On a number of occasions it is interesting to consider the analysis for $\lambda = 1$ separated from that for $\lambda < 1$.

Excess MSE for $\lambda = 1$

By applying in equation (5.49) the results of equations (5.36) and (5.19), and considering that

$$\xi_{\mathrm{min,p}} = \left(1 - 2\frac{N+1}{k+1}\right)\xi_{\mathrm{min}} = \left(1 - 2\frac{N+1}{k+1}\right)\sigma_n^2$$

for $\lambda = 1$ (see equations (5.42) and (5.47)), we can infer that

$$\begin{aligned}\xi(k) &= \left[1 - 2\frac{N+1}{k+1}\right]\sigma_n^2 + \mathrm{tr}\left\{\mathbf{R}E[\mathbf{S}_D(k)]\right\}\sigma_n^2 \\ &= \left[1 - 2\frac{N+1}{k+1} + \mathrm{tr}(\mathbf{R}\frac{\mathbf{R}^{-1}}{k+1})\right]\sigma_n^2 \text{ for } k \to \infty \\ &= \left(1 - 2\frac{N+1}{k+1} + \frac{N+1}{k+1}\right)\sigma_n^2 \text{ for } k \to \infty \\ &= \left(1 - \frac{N+1}{k+1}\right)\sigma_n^2 \text{ for } k \to \infty \end{aligned}$$

As can be noted the minimum MSE can be reached only after the algorithm has operated on a number of samples larger than the filter order.

Excess MSE for $\lambda < 1$

Again assuming that the mean-square error surface is quadratic as considered in equation (5.48), the expected excess in the MSE is then defined by

$$\Delta\xi(k) = E[\Delta\mathbf{w}^T(k)\mathbf{R}\Delta\mathbf{w}(k)] \tag{5.50}$$

The objective now is to calculate and analyze the excess MSE when $\lambda < 1$. From equation (5.30) one can show that

$$\Delta\mathbf{w}(k) = \lambda\mathbf{S}_D(k)\mathbf{R}_D(k-1)\Delta\mathbf{w}(k-1) + \mathbf{S}_D(k)\mathbf{x}(k)e_o(k) \tag{5.51}$$

By applying equation (5.51) to (5.50), it follows that

$$E[\Delta\mathbf{w}^T(k)\mathbf{R}\Delta\mathbf{w}(k)] = \rho_1 + \rho_2 + \rho_3 + \rho_4 \tag{5.52}$$

where

$$\begin{aligned}
\rho_1 &= \lambda^2 E[\Delta\mathbf{w}^T(k-1)\mathbf{R}_D(k-1)\mathbf{S}_D(k)\mathbf{R}\mathbf{S}_D(k)\mathbf{R}_D(k-1)\Delta\mathbf{w}(k-1)]\\
\rho_2 &= \lambda E[\Delta\mathbf{w}^T(k-1)\mathbf{R}_D(k-1)\mathbf{S}_D(k)\mathbf{R}\mathbf{S}_D(k)\mathbf{x}(k)e_o(k)]\\
\rho_3 &= \lambda E[\mathbf{x}^T(k)\mathbf{S}_D(k)\mathbf{R}\mathbf{S}_D(k)\mathbf{R}_D(k-1)\Delta\mathbf{w}(k-1)e_o(k)]\\
\rho_4 &= E[\mathbf{x}^T(k)\mathbf{S}_D(k)\mathbf{R}\mathbf{S}_D(k)\mathbf{x}(k)e_o^2(k)]
\end{aligned}$$

Now each term in equation (5.52) will be evaluated separately.

1- Evaluation of ρ_1

First note that as $k \to \infty$, it can be assumed that $\mathbf{R}_D(k) \approx \mathbf{R}_D(k-1)$, then

$$\rho_1 \approx \lambda^2 E[\Delta\mathbf{w}^T(k-1)\mathbf{R}\Delta\mathbf{w}(k-1)] \tag{5.53}$$

2- Evaluation of ρ_2

Since each element of $\mathbf{R}_D(k)$ is given by

$$r_{d,ij}(k) = \sum_{l=0}^{k} \lambda^{k-l} x(l-i)x(l-j) \tag{5.54}$$

for $0 \leq i,\ j \leq N$. Therefore,

$$E[r_{d,ij}(k)] = \sum_{l=0}^{k} \lambda^{k-l} E[x(l-i)x(l-j)]$$

If $x(k)$ is stationary, $r(i-j) = E[x(l-i)x(l-j)]$ is independent of the value l, then

$$E[r_{d,ij}(k)] = r(i-j)\frac{1-\lambda^{k+1}}{1-\lambda} \approx \frac{r(i-j)}{1-\lambda} \tag{5.55}$$

Equation (5.55) allows the conclusion that

$$E[\mathbf{R}_D(k)] \approx \frac{1}{1-\lambda}E[\mathbf{x}(k)\mathbf{x}^T(k)] = \frac{1}{1-\lambda}\mathbf{R} \tag{5.56}$$

In each step, it can be considered that

$$\mathbf{R}_D(k) = \frac{1}{1-\lambda}\mathbf{R} + \Delta\mathbf{R}(k) \tag{5.57}$$

where $\Delta\mathbf{R}(k)$ is a symmetric error matrix with zero-mean stochastic entries that are independent of the input signal. From equations (5.56) and (5.57), it can be concluded that

$$\mathbf{S}_D(k)\mathbf{R} \approx (1-\lambda)\left[\mathbf{I} - (1-\lambda)\mathbf{R}^{-1}\Delta\mathbf{R}(k)\right] \tag{5.58}$$

where in the last relation $\mathbf{S}_D(k)\Delta\mathbf{R}(k)$ was considered approximately equal to

$$(1-\lambda)\mathbf{R}^{-1}\Delta\mathbf{R}(k)$$

by using equation (5.56) and disregarding second-order errors.

In the long run, it is known that $E[\mathbf{S}_D(k)\mathbf{R}] = (1-\lambda)\mathbf{I}$, that means the second term inside the square bracket in equation (5.58) is a measure of the perturbation caused by $\Delta\mathbf{R}(k)$ in the product $\mathbf{S}_D(k)\mathbf{R}$. Denoting the perturbation by $\Delta\mathbf{I}(k)$, that is

$$\Delta\mathbf{I}(k) = (1-\lambda)\mathbf{R}^{-1}\Delta\mathbf{R}(k) \tag{5.59}$$

it can be concluded that

$$\begin{aligned} \rho_2 &\approx \lambda(1-\lambda)E\left\{\Delta\mathbf{w}^T(k-1)[\mathbf{I} - \Delta\mathbf{I}^T(k)]\mathbf{x}(k)e_o(k)\right\} \\ &\approx \lambda(1-\lambda)E[\Delta\mathbf{w}^T(k-1)]E[\mathbf{x}(k)e_o(k)] = 0 \end{aligned} \tag{5.60}$$

where it was considered that $\Delta\mathbf{w}^T(k-1)$ is independent of $\mathbf{x}(k)$ and $e_o(k)$, $\Delta\mathbf{I}(k)$ was also considered an independent error matrix with zero mean, and finally we used the fact that $\mathbf{x}(k)$ and $e_o(k)$ are orthogonal.

3- Following a similar approach it can be shown that

$$\begin{aligned} \rho_3 &\approx \lambda(1-\lambda)E\left\{\mathbf{x}^T(k)[\mathbf{I} - \Delta\mathbf{I}(k)]\Delta\mathbf{w}(k-1)e_o(k)\right\} \\ &\approx \lambda(1-\lambda)E[\mathbf{x}^T(k)e_o(k)]E[\Delta\mathbf{w}(k-1)] = 0 \end{aligned} \tag{5.61}$$

4- Evaluation of ρ_4

$$\begin{aligned} \rho_4 &= E[\mathbf{x}^T(k)\mathbf{S}_D(k)\mathbf{R}\mathbf{S}_D(k)\mathbf{R}\mathbf{R}^{-1}\mathbf{x}(k)e_o^2(k)] \\ &\approx (1-\lambda)^2E\left\{\mathbf{x}^T(k)[\mathbf{I} - \Delta\mathbf{I}(k)]^2\mathbf{R}^{-1}\mathbf{x}(k)\right\}\xi_{\min} \end{aligned} \tag{5.62}$$

where equations (5.58) and (5.29) were used and $e_o(k)$ was considered independent of $\mathbf{x}(k)$ and $\Delta\mathbf{I}(k)$. By using the property that

$$E\left\{\mathbf{x}^T(k)[\mathbf{I} - \Delta\mathbf{I}(k)]^2\mathbf{R}^{-1}\mathbf{x}(k)\right\} = \text{tr}E\left\{[\mathbf{I} - \Delta\mathbf{I}(k)]^2\mathbf{R}^{-1}\mathbf{x}(k)\mathbf{x}^T(k)\right\}$$

and recalling that $\Delta\mathbf{I}(k)$ has zero mean and is independent of $\mathbf{x}(k)$, then equation (5.62) is simplified to

$$\rho_4 = (1-\lambda)^2\text{tr}\{\mathbf{I} + E[\Delta\mathbf{I}^2(k)]\}\xi_{\min} \tag{5.63}$$

where $\mathrm{tr}[\cdot]$ means trace of $[\cdot]$, and we utilized the fact that $E\{\mathbf{R}^{-1}\mathbf{x}(k)\mathbf{x}^T(k)\} = \mathbf{I}$.

By using equations (5.53), (5.60), and (5.63), it follows that

$$E[\Delta\mathbf{w}^T(k)\mathbf{R}\Delta\mathbf{w}(k)] = \lambda^2 E[\Delta\mathbf{w}^T(k-1)\mathbf{R}\Delta\mathbf{w}(k-1)] + (1-\lambda)^2 \mathrm{tr}\{\mathbf{I} + E[\Delta\mathbf{I}^2(k)]\}\xi_{\min} \tag{5.64}$$

Asymptotically, the solution of the above equation is

$$\xi_{\mathrm{exc}} = \frac{1-\lambda}{1+\lambda}\mathrm{tr}\{\mathbf{I} + E[\Delta\mathbf{I}^2(k)]\}\xi_{\min} \tag{5.65}$$

Note that the term given by $E[\Delta\mathbf{I}^2(k)]$ is not easy to estimate and is dependent on fourth-order statistics of the input signal. However, in specific situations, it is possible to compute an approximate estimate for this matrix. In steady state, it can be considered for white noise input signal that only the diagonal elements of **R** and Δ**R** are important to the generation of excess MSE. Even when the input signal is not white, this diagonal dominance can be considered a reasonable approximation in most of the cases. From the definition of $\Delta\mathbf{I}(k)$ in equation (5.59), it follows that

$$E[\Delta\mathbf{I}_{ii}^2(k)] = (1-\lambda)^2 \frac{E[\Delta r_{ii}^2(k)]}{[\sigma_x^2]^2} \tag{5.66}$$

where σ_x^2 is variance of $x(k)$. By calculating $\Delta\mathbf{R}(k) - \lambda\Delta\mathbf{R}(k-1)$ using equation (5.57), we show that

$$\Delta r_{ii}(k) = \lambda\Delta r_{ii}(k-1) + x(k-i)x(k-i) - r_{ii} \tag{5.67}$$

Squaring the above equation, applying the expectation operation, and using the independence between $\Delta r_{ii}(k)$ and $x(k)$, it follows that

$$E[\Delta r_{ii}^2(k)] = \lambda^2 E[\Delta r_{ii}^2(k-1)] + E\left\{[x(k-i)x(k-i) - r_{ii}]^2\right\} \tag{5.68}$$

Therefore, asymptotically

$$E[\Delta r_{ii}^2(k)] = \frac{1}{1-\lambda^2}\sigma_{x^2(k-i)}^2 = \frac{1}{1-\lambda^2}\sigma_{x^2}^2 \tag{5.69}$$

By substituting equation (5.69) in (5.66), it becomes

$$E[\Delta\mathbf{I}_{ii}^2(k)] = \frac{1-\lambda}{1+\lambda}\frac{\sigma_{x^2}^2}{(\sigma_x^2)^2} = \frac{1-\lambda}{1+\lambda}\mathcal{K} \tag{5.70}$$

where $\mathcal{K} = \frac{\sigma_{x^2}^2}{(\sigma_x^2)^2}$ is dependent on input signal statistics. For Gaussian signals, $\mathcal{K} = 2$ [7].

Returning to our main objective, the excess MSE can then be described as

$$\xi_{\mathrm{exc}} = (N+1)\frac{1-\lambda}{1+\lambda}\left(1 + \frac{1-\lambda}{1+\lambda}\mathcal{K}\right)\xi_{\min} \tag{5.71}$$

If λ is approximately one and $\mathcal{K}$ is not very large then

$$\xi_{\text{exc}} = (N+1)\frac{1-\lambda}{1+\lambda}\xi_{\min} \tag{5.72}$$

this expression can be reached through a simpler analysis [6]. However, the more complete derivation shown here can give more insight to the interpretation of the results obtained by using the RLS algorithm, mainly when λ is not very close to one.

The misadjustment formula can be deduced from equation (5.71)

$$M = \frac{\xi_{\text{exc}}}{\xi_{\min}} = (N+1)\frac{1-\lambda}{1+\lambda}\left(1+\frac{1-\lambda}{1+\lambda}\mathcal{K}\right) \tag{5.73}$$

As can be noted, the decrease of λ from one brings a fourth-order statistics term into the picture by increasing the misadjustment. Then, the fast adaptation of the RLS algorithm, that corresponds to smaller λ, brings a noisier steady-state response. Therefore, when working in a stationary environment the best choice for λ would be one, if the excess MSE in the steady state is considered high for other values of λ. However, other problems such as instability due to quantization noise are prone to occur when $\lambda = 1$.

5.4 BEHAVIOR IN NONSTATIONARY ENVIRONMENTS

In cases where the input signal and/or the desired signal are nonstationary, the optimal values of the coefficients are time variant and described by $\mathbf{w}_o(k)$. That means the autocorrelation matrix $\mathbf{R}(k)$ and/or the cross-correlation vector $\mathbf{p}(k)$ are time variant. For example, typically in a system identification application the autocorrelation matrix $\mathbf{R}(k)$ is time invariant while the cross-correlation matrix $\mathbf{p}(k)$ is time variant, because in this case the designer can choose the input signal. On the other hand, in equalization, prediction, and signal enhancement applications both the input and the desired signal are nonstationary leading to time-varying matrices $\mathbf{R}(k)$ and $\mathbf{p}(k)$.

The objective in the present section is to analyze how close the RLS algorithm is able to track the time-varying solution $\mathbf{w}_o(k)$. Also, it is of interest to learn how the tracking error in $\mathbf{w}(k)$ affects the output MSE [7]. Here, the effects of the measurement noise are not considered, since only the nonstationary effects are desired. Also, both effects on the MSE can be added since, in general, they are independent.

Recall from equations (5.8) and (5.9) that

$$\mathbf{w}(k) = \mathbf{w}(k-1) + \mathbf{S}_D(k)\mathbf{x}(k)[d(k) - \mathbf{x}^T(k)\mathbf{w}(k-1)] \tag{5.74}$$

and

$$d(k) = \mathbf{x}^T(k)\mathbf{w}_o(k-1) + e'_o(k) \tag{5.75}$$

The error signal $e'_o(k)$ is the minimum error at iteration k being generated by the nonstationarity of the environment. One can replace equation (5.75) in (5.74) in order to obtain the following relation

$$\mathbf{w}(k) = \mathbf{w}(k-1) + \mathbf{S}_D(k)\mathbf{x}(k)\mathbf{x}^T(k)[\mathbf{w}_o(k-1) - \mathbf{w}(k-1)] + \mathbf{S}_D(k)\mathbf{x}(k)e'_o(k) \tag{5.76}$$

By taking the expected value of equation (5.76), considering that $\mathbf{x}(k)$ and $e'_o(k)$ are approximately orthogonal, and that $\mathbf{w}(k-1)$ is independent of $\mathbf{x}(k)$, then

$$E[\mathbf{w}(k)] = E[\mathbf{w}(k-1)] + E[\mathbf{S}_D(k)\mathbf{x}(k)\mathbf{x}^T(k)]\left\{\mathbf{w}_o(k-1) - E[\mathbf{w}(k-1)]\right\} \tag{5.77}$$

It is now needed to compute $E[\mathbf{S}_D(k)\mathbf{x}(k)\mathbf{x}^T(k)]$ in the case of nonstationary input signal. From equations (5.54) and (5.56), one can show that

$$\mathbf{R}_D(k) = \sum_{l=0}^{k} \lambda^{k-l}\mathbf{R}(l) + \Delta\mathbf{R}(k) \tag{5.78}$$

since $E[\mathbf{R}_D(k)] = \sum_{l=0}^{k}\lambda^{k-l}\mathbf{R}(l)$. The matrix $\Delta\mathbf{R}(k)$ is again considered a symmetric error matrix with zero-mean stochastic entries that are independent of the input signal.

If the environment is considered to be varying at a slower pace than the memory of the adaptive RLS algorithm, then

$$\mathbf{R}_D(k) \approx \frac{1}{1-\lambda}\mathbf{R}(k) + \Delta\mathbf{R}(k) \tag{5.79}$$

Considering that $(1-\lambda)||\mathbf{R}^{-1}(k)\Delta\mathbf{R}(k)|| < 1$ and using the same procedure to deduce equation (5.58), we obtain

$$\mathbf{S}_D(k) \approx (1-\lambda)\mathbf{R}^{-1}(k) - (1-\lambda)^2\mathbf{R}^{-1}(k)\Delta\mathbf{R}(k)\mathbf{R}^{-1}(k) \tag{5.80}$$

it then follows that

$$\begin{aligned} E[\mathbf{w}(k)] &= E[\mathbf{w}(k-1)] + \big\{(1-\lambda)E[\mathbf{R}^{-1}(k)\mathbf{x}(k)\mathbf{x}^T(k)] \\ &\quad - (1-\lambda)^2E[\mathbf{R}^{-1}(k)\Delta\mathbf{R}(k)\mathbf{R}^{-1}(k)\mathbf{x}(k)\mathbf{x}^T(k)]\big\}\left\{\mathbf{w}_o(k-1) - E[\mathbf{w}(k-1)]\right\} \\ &\approx E[\mathbf{w}(k-1)] + (1-\lambda)\left\{\mathbf{w}_o(k-1) - E[\mathbf{w}(k-1)]\right\} \end{aligned} \tag{5.81}$$

where it was considered that $\Delta\mathbf{R}(k)$ is independent of $\mathbf{x}(k)$ and has zero expected value.

Now defining the lag-error vector in the coefficients as

$$\mathbf{l_w}(k) = E[\mathbf{w}(k)] - \mathbf{w}_o(k) \tag{5.82}$$

From equation (5.81), it can be concluded that

$$\mathbf{l_w}(k) = \lambda\mathbf{l_w}(k-1) - \mathbf{w}_o(k) + \mathbf{w}_o(k-1) \tag{5.83}$$

Equation (5.83) is equivalent to say that the lag is generated by applying the optimal instantaneous value $\mathbf{w}_o(k)$ through a first-order discrete-time filter as follows:

$$L_i(z) = -\frac{z-1}{z-\lambda}W_{oi}(z) \tag{5.84}$$

The discrete-time filter transient response converges with a time constant given by

$$\tau = \frac{1}{1-\lambda} \tag{5.85}$$

The time constant is of course the same for each individual coefficient. Note that the tracking ability of the coefficients in the RLS algorithm is independent of the eigenvalues of the input signal correlation matrix.

The lag in the coefficients leads to an excess MSE. In order to calculate the MSE suppose that the optimal coefficient values are first-order Markov processes described by

$$\mathbf{w}_o(k) = \lambda_{\mathbf{w}}\mathbf{w}_o(k-1) + \mathbf{n}_{\mathbf{w}}(k) \tag{5.86}$$

where $\mathbf{n}_{\mathbf{w}}(k)$ is a vector whose elements are zero-mean white noise processes with variance $\sigma^2_{\mathbf{w}}$, and $\lambda_{\mathbf{w}} < 1$. Note that $\lambda < \lambda_{\mathbf{w}} < 1$, since the optimal coefficients values must vary slower than the filter tracking speed, that means $\frac{1}{1-\lambda} < \frac{1}{1-\lambda_{\mathbf{w}}}$.

The excess MSE due to lag is then given by (see the derivations around equation (3.41))

$$\begin{aligned}
\xi_{\text{lag}} &= E[\mathbf{l}^T_{\mathbf{w}}(k)\mathbf{R}\mathbf{l}_{\mathbf{w}}(k)] \\
&= E\left\{\text{tr}[\mathbf{R}\mathbf{l}_{\mathbf{w}}(k)\mathbf{l}^T_{\mathbf{w}}(k)]\right\} \\
&= \text{tr}\left\{\mathbf{R}E[\mathbf{l}_{\mathbf{w}}(k)\mathbf{l}^T_{\mathbf{w}}(k)]\right\} \\
&= \text{tr}\left\{\mathbf{\Lambda}E[\mathbf{l}'_{\mathbf{w}}(k)\mathbf{l}'^T_{\mathbf{w}}(k)]\right\} \\
&= \sum_{i=0}^{N} \lambda_i E[l'^2_i(k)]
\end{aligned} \tag{5.87}$$

For $\lambda_{\mathbf{w}}$ not close to one, it is a bit more complicated to deduce the excess MSE due to lag than for $\lambda_{\mathbf{w}} \approx 1$. However, the effort is worth it because the resulting expression is more accurate. From equation (5.84), we can see 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. On the other hand, the coefficients of the unknown system are generated by applying each element of the noise vector $\mathbf{n}_{\mathbf{w}}(k)$ 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 the elements of the noise vector $\mathbf{n}_{\mathbf{w}}(k)$ to a discrete-time filter with transfer function

$$H(z) = \frac{-(z-1)z}{(z-\lambda)(z-\lambda_{\mathbf{w}})} \tag{5.88}$$

This transfer function consists of a cascade of the lag filter with the all-pole filter representing the first-order Markov process. The solution for the variance of the lag terms l_i can be computed through the inverse $\mathcal{Z}$-transform as follows:

$$E[l'^2_i(k)] = \frac{1}{2\pi\jmath}\oint H(z)H(z^{-1})\sigma^2_{\mathbf{w}}z^{-1}\,dz \tag{5.89}$$

The above integral can be solved using the residue theorem as previously shown in the LMS algorithm case.

Using the solution for the variance of the lag terms of equation (5.89) for values of $\lambda_{\mathbf{w}} < 1$, and substituting the result in the last term of equation (5.87) it can be shown that

$$\begin{aligned}\xi_{\text{lag}} &\approx \frac{\text{tr}[\mathbf{R}]\sigma_{\mathbf{w}}^2}{\lambda_{\mathbf{w}}(1+\lambda^2)-\lambda(1+\lambda_{\mathbf{w}}^2)}\left(\frac{1-\lambda}{1+\lambda}-\frac{1-\lambda_{\mathbf{w}}}{1+\lambda_{\mathbf{w}}}\right)\\ &= \frac{(N+1)\sigma_{\mathbf{w}}^2\sigma_x^2}{\lambda_{\mathbf{w}}(1+\lambda^2)-\lambda(1+\lambda_{\mathbf{w}}^2)}\left(\frac{1-\lambda}{1+\lambda}-\frac{1-\lambda_{\mathbf{w}}}{1+\lambda_{\mathbf{w}}}\right)\end{aligned} \tag{5.90}$$

where it was used the fact that $\text{tr}[\mathbf{R}] = \sum_{i=0}^{N}\lambda_i = (N+1)\sigma_x^2$, for a tap delay line. It should be noticed that assumptions such as the correlation matrix **R** being diagonal and the input signal being white noise were not required in this derivation.

If $\lambda = 1$ and $\lambda_{\mathbf{w}} \approx 1$, the MSE due to lag tends to infinity indicating that the RLS algorithm in this case cannot track any change in the environment. On the other hand, for $\lambda < 1$ the algorithm can track variations in the environment, leading to an excess MSE that depends on the variance of the optimal coefficient disturbance and on the input signal variance.

For $\lambda_{\mathbf{w}} = 1$ and $\lambda \approx 1$, it is possible to rewrite equation (5.90) as

$$\xi_{\text{lag}} \approx (N+1)\frac{\sigma_{\mathbf{w}}^2}{2(1-\lambda)}\sigma_x^2 \tag{5.91}$$

The total excess MSE accounting for the lag and finite memory is given by

$$\xi_{\text{total}} \approx (N+1)\left[\frac{1-\lambda}{1+\lambda}\xi_{\text{min}} + \frac{\sigma_{\mathbf{w}}^2\sigma_x^2}{2(1-\lambda)}\right] \tag{5.92}$$

By differentiating the above equation with respect to λ and setting the result to zero, an optimum value for λ can be found that yields minimum excess MSE.

$$\lambda_{\text{opt}} = \frac{1-\frac{\sigma_{\mathbf{w}}\sigma_x}{2\sigma_n}}{1+\frac{\sigma_{\mathbf{w}}\sigma_x}{2\sigma_n}} \tag{5.93}$$

In the above equation we used $\sigma_n = \sqrt{\xi_{\text{min}}}$. Note that the optimal value of λ does not depend on the adaptive-filter order N, and can be used when it falls in an acceptable range of values for λ. Also, this value is optimum only when quantization effects are not important and the first-order Markov model (with $\lambda_{\mathbf{w}} \approx 1$) is a good approximation for the nonstationarity of the desired signal.

When implemented with finite-precision arithmetic, the conventional RLS algorithm behavior can differ significantly from what is expected under infinite precision. A series of inconvenient effects can show up in the practical implementation of the conventional RLS algorithm, such as divergence and freezing in the updating of the adaptive-filter coefficients. Appendix C, presents a detailed analysis of the finite-wordlength effects in the RLS algorithm.

5.5 COMPLEX RLS ALGORITHM

In the complex data case the RLS objective function is given by

$$\begin{aligned}\xi^d(k) &= \sum_{i=0}^{k} \lambda^{k-i}|\varepsilon(i)|^2 = \sum_{i=0}^{k} \lambda^{k-i}|d(i) - \mathbf{w}^H(i)\mathbf{x}(k)|^2 \\ &= \sum_{i=0}^{k} \lambda^{k-i}\left[d(i) - \mathbf{w}^H(i)\mathbf{x}(k)\right]\left[d^*(i) - \mathbf{w}^T(i)\mathbf{x}^*(k)\right] \qquad (5.94)\end{aligned}$$

Differentiating $\xi^d(k)$ with respect to the complex coefficient $\mathbf{w}^*(k)$ leads to[3]

$$\frac{\partial \xi^d(k)}{\partial \mathbf{w}^*(k)} = -\sum_{i=0}^{k} \lambda^{k-i}\mathbf{x}(i)[d^*(i) - \mathbf{w}^T(i)\mathbf{x}^*(k)] \qquad (5.95)$$

The optimal vector $\mathbf{w}(k)$ that minimizes the least-squares error, is computed by equating the above equation to zero that is

$$-\sum_{i=0}^{k} \lambda^{k-i}\mathbf{x}(i)\mathbf{x}^H(i)\mathbf{w}(k) + \sum_{i=0}^{k} \lambda^{k-i}\mathbf{x}(i)d^*(i) = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

leading to the following expression

$$\begin{aligned}\mathbf{w}(k) &= \left[\sum_{i=0}^{k} \lambda^{k-i}\mathbf{x}(i)\mathbf{x}^H(i)\right]^{-1} \sum_{i=0}^{k} \lambda^{k-i}\mathbf{x}(i)d^*(i) \\ &= \mathbf{R}_D^{-1}(k)\mathbf{p}_D(k) \qquad (5.96)\end{aligned}$$

The matrix inversion lemma to the case of complex data is given by

$$\mathbf{S}_D(k) = \mathbf{R}_D^{-1}(k) = \frac{1}{\lambda}\left[\mathbf{S}_D(k-1) - \frac{\mathbf{S}_D(k-1)\mathbf{x}(k)\mathbf{x}^H(k)\mathbf{S}_D(k-1)}{\lambda + \mathbf{x}^H(k)\mathbf{S}_D(k-1)\mathbf{x}(k)}\right] \qquad (5.97)$$

The complete conventional RLS algorithm is described in Algorithm 5.3.

An alternative complex RLS algorithm has an updating equation described by

$$\mathbf{w}(k) = \mathbf{w}(k-1) + e^*(k)\mathbf{S}_D(k)\mathbf{x}(k) \qquad (5.98)$$

where

$$e(k) = d(k) - \mathbf{w}^H(k-1)\mathbf{x}(k) \qquad (5.99)$$

With equation (5.98), it is straightforward to generate an alternative conventional RLS algorithm as shown in Algorithm 5.4.

[3] Again the reader should recall that when computing the gradient with respect to $\mathbf{w}^*(k)$, $\mathbf{w}(k)$ is treated as a constant.

Algorithm 5.3

Conventional Complex RLS Algorithm

```
Initialization
  $\mathbf{S}_D(-1) = \delta\mathbf{I}$
where $\delta$ can be the inverse of the input signal power estimate
  $\mathbf{p}_D(-1) = \mathbf{x}(-1) = [0\,0\ldots 0]^T$
Do for $k \geq 0$ :
   $\mathbf{S}_D(k) = \frac{1}{\lambda}[\mathbf{S}_D(k-1) - \frac{\mathbf{S}_D(k-1)\mathbf{x}(k)\mathbf{x}^H(k)\mathbf{S}_D(k-1)}{\lambda+\mathbf{x}^H(k)\mathbf{S}_D(k-1)\mathbf{x}(k)}]$
   $\mathbf{p}_D(k) = \lambda\mathbf{p}_D(k-1) + d^*(k)\mathbf{x}(k)$
   $\mathbf{w}(k) = \mathbf{S}_D(k)\mathbf{p}_D(k)$
If necessary compute
   $y(k) = \mathbf{w}^H(k)\mathbf{x}(k)$
   $\varepsilon(k) = d(k) - y(k)$
```

Algorithm 5.4

Alternative Complex RLS Algorithm

```
Initialization
  $\mathbf{S}_D(-1) = \delta\mathbf{I}$
where $\delta$ can be the inverse of an estimate of the input signal power
  $\mathbf{x}(-1) = \mathbf{w}(-1) = [0\,0\ldots 0]^T$
Do for $k \geq 0$
   $e(k) = d(k) - \mathbf{w}^H(k-1)\mathbf{x}(k)$
   $\boldsymbol{\psi}(k) = \mathbf{S}_D(k-1)\mathbf{x}(k)$
   $\mathbf{S}_D(k) = \frac{1}{\lambda}[\mathbf{S}_D(k-1) - \frac{\boldsymbol{\psi}(k)\boldsymbol{\psi}^H(k)}{\lambda+\boldsymbol{\psi}^H(k)\mathbf{x}(k)}]$
   $\mathbf{w}(k) = \mathbf{w}(k-1) + e^*(k)\mathbf{S}_D(k)\mathbf{x}(k)$
If necessary compute
   $y(k) = \mathbf{w}^H(k)\mathbf{x}(k)$
   $\varepsilon(k) = d(k) - y(k)$
```

5.6 SIMULATION EXAMPLES

In this section, some adaptive-filtering problems described in the last two chapters are solved using the conventional RLS algorithm presented in this chapter.

Example 5.3: System Identification Simulations

The conventional RLS algorithm is employed in the identification of the system described in the subsection 3.6.2. The forgetting factor is chosen as $\lambda = 0.99$.

Solution:

In the first test, we address the sensitivity of the RLS algorithm to the eigenvalue spread of the input signal correlation matrix. The measured simulation results are obtained by ensemble averaging 200 independent runs. The learning curves of the mean-squared *a priori* error are depicted in Fig. 5.4, for different values of the eigenvalue spread. Also, the measured misadjustment in each example is given in Table 5.1. From these results, we conclude that the RLS algorithm is insensitive to the eigenvalue spread. It is worth mentioning at this point that the convergence speed of the RLS algorithm is affected by the choice of λ, since a smaller value of λ leads to faster convergence while increasing the misadjustment in stationary environment. Table 5.1 shows the misadjustment predicted by theory, calculated using the relation repeated below. As can be seen from this table the analytical results agree with those obtained through simulations.

$$M = (N+1)\frac{1-\lambda}{1+\lambda}(1 + \frac{1-\lambda}{1+\lambda}\mathcal{K})$$

Table 5.1 Evaluation of the RLS Algorithm

	Misadjustment	
$\frac{\lambda_{\max}}{\lambda_{\min}}$	**Experiment**	**Theory**
1	0.04211	0.04020
20	0.04211	0.04020
80	0.04547	0.04020

The conventional RLS algorithm is implemented with finite-precision arithmetic, using fixed-point representation with 16, 12, and 10 bits, respectively. The results presented are measured before any sign of instability is noticed. Table 5.2 summarizes the results of the finite-precision implementation of the conventional RLS algorithm. Note that in most cases there is a close agreement between the measurement results and those predicted by the equations given below. These equations correspond to equations (C.37) and (C.48) derived in Appendix C.

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

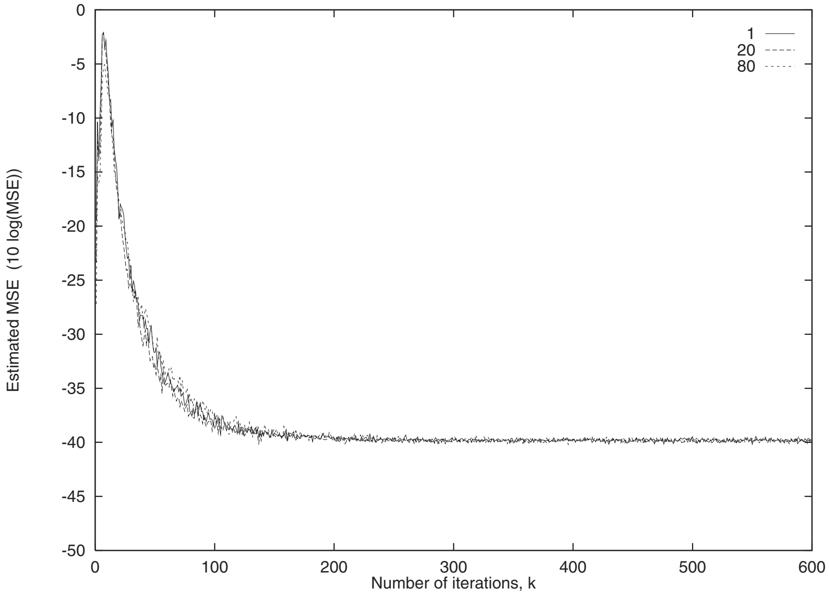


Figure 5.4 Learning curves for RLS algorithm for eigenvalue spreads: 1, 20, and 80; $\lambda = 0.99$.

$$\xi(k)_Q \approx \xi_{\min} + \sigma_e^2 + \frac{(N+1)\sigma_{\mathbf{w}}^2\sigma_x^2}{2\lambda(1-\lambda)}$$

For the simulations with 12 and 10 bits, the discrepancy between the measured and theoretical estimates of $E[||\Delta\mathbf{w}(k)_Q||^2]$ are caused by the freezing of some coefficients.

If the results presented here are compared with the results presented in Table 3.2 for the LMS, we notice that both the LMS and the RLS algorithms performed well in the finite-precision implementation. The reader should bear in mind that the conventional RLS algorithm requires an expensive strategy to keep the deterministic correlation matrix positive definite, as discussed in Appendix C.

Table 5.2 Results of the Finite-Precision Implementation of the RLS Algorithm

	$\xi(k)_Q$		$E[\|\|\Delta\mathbf{w}(k)_Q\|\|^2]$	
No. of bits	**Experiment**	**Theory**	**Experiment**	**Theory**
16	1.566 10^{-3}	1.500 10^{-3}	6.013 10^{-5}	6.061 10^{-5}
12	1.522 10^{-3}	1.502 10^{-3}	3.128 10^{-5}	6.261 10^{-5}
10	1.566 10^{-3}	1.532 10^{-3}	6.979 10^{-5}	9.272 10^{-5}

The simulations related to the experiment described for nonstationary environments are also performed. From the simulations we measure the total excess MSE, and then compare the results to those obtained with the expression below.

$$\xi_{\text{exc}} \approx (N+1)\frac{1-\lambda}{1+\lambda}(1+\frac{1-\lambda}{1+\lambda}\mathcal{K})\xi_{\min} + \frac{(N+1)\sigma_{\mathbf{w}}^2\sigma_x^2}{\lambda_{\mathbf{w}}(1+\lambda^2)-\lambda(1+\lambda_{\mathbf{w}}^2)}(\frac{1-\lambda}{1+\lambda}-\frac{1-\lambda_{\mathbf{w}}}{1+\lambda_{\mathbf{w}}})$$

An attempt to use the optimal value of λ is made. The predicted optimal value, in this case, is too small and as a consequence $\lambda = 0.99$ is used. The measured excess MSE is 0.0254, whereas the theoretical value predicted by the above equation is 0.0418. Note that the theoretical result is not as accurate as all the previous cases discussed so far, due to a number of approximations used in the analysis. However, the above equation provides a good indication of what is expected in the practical implementation. By choosing a smaller value for λ a better tracking performance is obtained, a situation where the above equation is not as accurate.

□

Example 5.4: Signal Enhancement Simulations

We solved the same signal enhancement problem described in the subsection 4.7.1 with the conventional RLS and LMS algorithms.

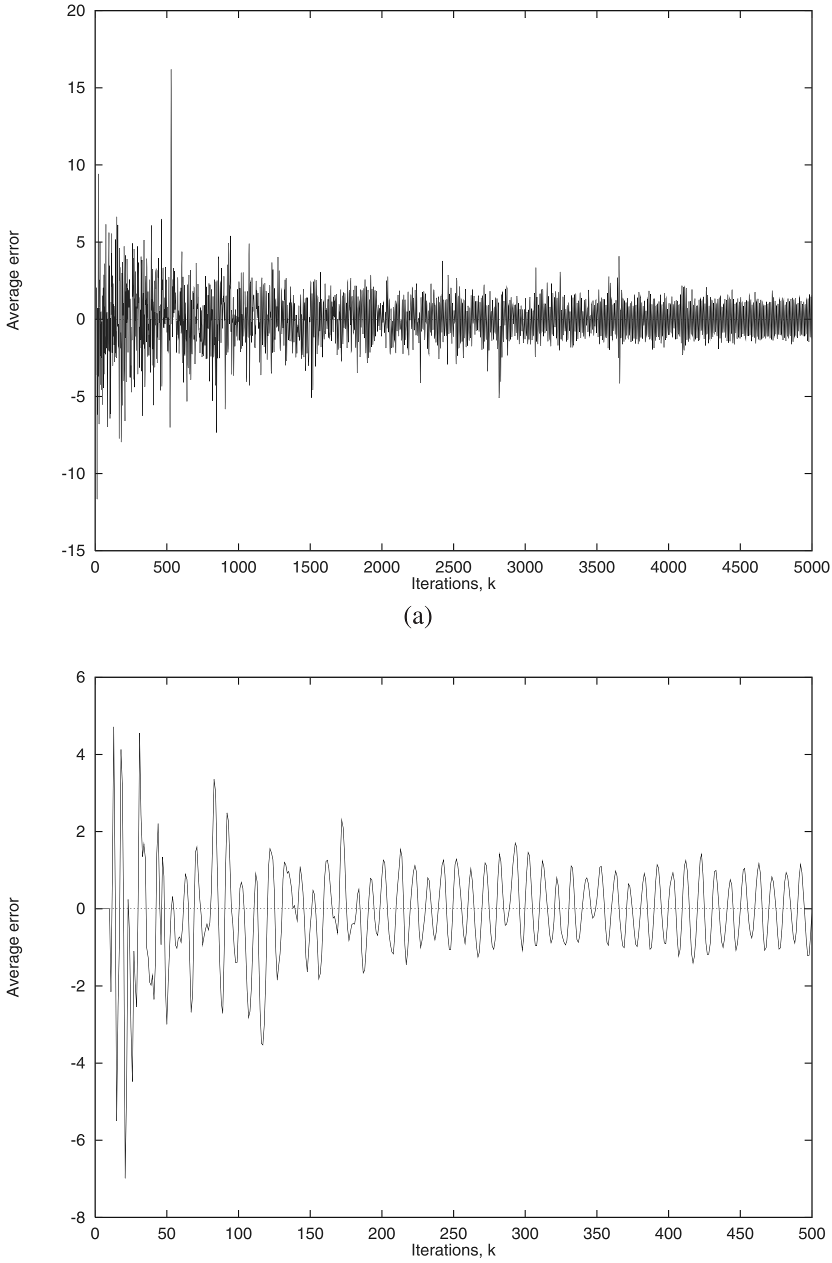
Solution:

For the LMS algorithm, the convergence factor is chosen $\mu_{\max}/5$. The resulting value for μ in the LMS case is 0.001, whereas $\lambda = 1.0$ is used for the RLS algorithm. The learning curves for the algorithms are shown in Fig. 5.5, where we can verify the faster convergence of the RLS algorithm. By plotting the output errors after convergence, we noted the large variance of the MSE for both algorithms. This result is due to the small signal to noise ratio, in this case. Fig. 5.6 depicts the output error and its DFT with 128 points for the RLS algorithm. In both cases, we can clearly detect the presence of the sinusoid.

□

5.7 CONCLUDING REMARKS

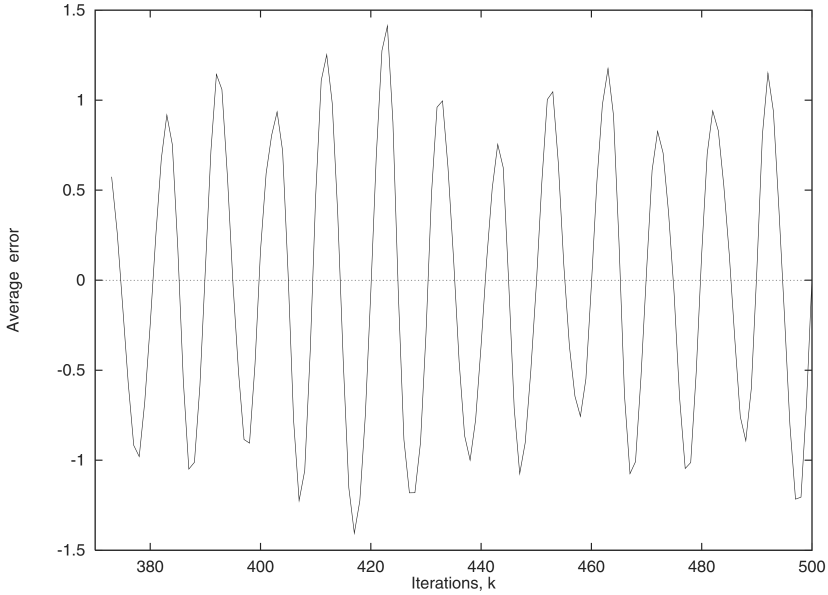
In this chapter, we introduced the conventional RLS algorithm and discussed various aspects related to its performance behavior. Much of the results obtained herein through mathematical analysis are valid for the whole class of RLS algorithms to be presented in the following chapters, except for the finite-precision analysis since that depends on the form the internal calculations of each algorithm are performed. The analysis presented here is far from being complete. However, the main aspects of the conventional RLS have been addressed, such as: convergence behavior and tracking capabilities.



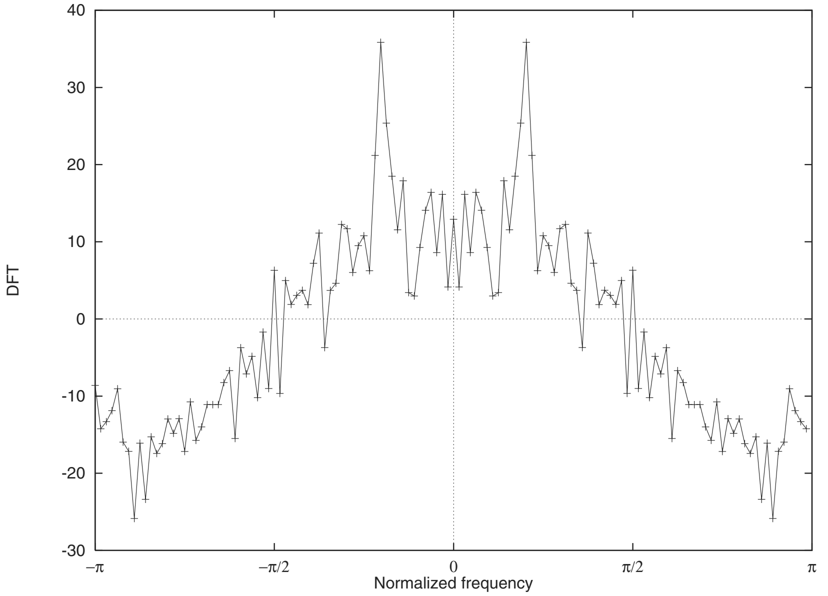
(a)

(b)

Figure 5.5 Learning curves for the (a) LMS and (b) RLS algorithms.



(a)



(b)

Figure 5.6 (a) Output error for the RLS algorithm and (b) DFT of the output error.

The interested reader should consult [9]-[11] for some further results. Appendix C complements this chapter by addressing the finite-precision analysis of the conventional RLS algorithm.

From the analysis presented, one can conclude that the computational complexity and the stability in finite-precision implementations are two aspects to be concerned. When the elements of the input signal vector consist of delayed versions of the same signal, it is possible to derive a number of fast RLS algorithms whose computational complexity is of order N per output sample. Several different classes of these algorithms are presented in the following chapters. In all cases, their stability conditions in finite-precision implementation are briefly discussed.

For the general case where the elements of the input signal vector have different origins the QR-RLS algorithm is a good alternative to the conventional RLS algorithm. The stability of the QR-RLS algorithm can be easily guaranteed.

The conventional RLS algorithm is fully tested in a number of simulation results included in this chapter. These examples were meant to verify the theoretical results discussed in the present chapter and to compare the RLS algorithm with the LMS algorithm.

The LMS algorithm is usually referred to as stochastic gradient algorithm originated from the stochastic formulation of the Wiener filter which in turn deals with stationary noises and signals. The RLS algorithm is derived from a deterministic formulation meant to achieve weighted least-squares error minimization in a sequential recursive format. A widely known generalization of the Wiener filter is the Kalman filter which deals with nonstationary noises and signals utilizing a stochastic formulation. However, it is possible to show that the discrete-time version of the Kalman filtering algorithm can be considered to be a generalization of the RLS algorithm. In Appendix D we present a brief description of Kalman filters as well as its relationship with the RLS algorithm.

5.8 REFERENCES

1. G. C. Goodwin and R. L. Payne, *Dynamic System Identification: Experiment Design and Data Analysis*, Academic Press, New York, NY, 1977.

2. S. Haykin, *Adaptive Filter Theory*, Prentice Hall, Englewood Cliffs, NJ, 4th edition, 2002.

3. S. H. Ardalan, "Floating-point analysis of recursive least-squares and least-mean squares adaptive filters," *IEEE Trans. on Circuits and Systems*, vol. CAS-33, pp. 1192-1208, Dec. 1986.

4. J. M. Cioffi, "Limited precision effects in adaptive filtering," *IEEE Trans. on Circuits and Systems*, vol. CAS-34, pp. 821-833, July 1987.

5. R. S. Medaugh and L. J. Griffiths, "A comparison of two linear predictors," *Proc. IEEE Intern. Conf. on Acoust., Speech, Signal Processing*, Atlanta, GA, pp. 293-296, April 1981.

6. F. Ling and J. G. Proakis, "Nonstationary learning characteristics of least squares adaptive estimation algorithms," *Proc. IEEE Intern. Conf. on Acoust., Speech, Signal Processing*, San Diego, CA, pp. 30.3.1.-30.3.4, March 1984.

7. E. Eleftheriou and D. D. Falconer, "Tracking properties and steady-state performance of RLS adaptive filter algorithms," *IEEE Trans. on Acoust., Speech, and Signal Processing*, vol. ASSP-34, pp. 1097-1110, Oct. 1986.

8. J. M. Cioffi and T. Kailath, "Fast recursive-least-squares transversal filters for adaptive filtering," *IEEE Trans. on Acoust., Speech, and Signal Processing*, vol. ASSP-32, pp. 304-337, April 1984.

9. S. Ardalan, "On the sensitivity of transversal RLS algorithms to random perturbations in the filter coefficients," *IEEE Trans. on Acoust., Speech, and Signal Processing,* vol. 36, pp. 1781-1783, Nov. 1988.

10. C. R. Johnson, Jr., *Lectures on Adaptive Parameter Estimation*, Prentice Hall, Englewood Cliffs, NJ, 1988.

11. O. M. Macchi and N. J. Bershad, "Adaptive recovery of a chirped sinusoid in noise, Part 1: performance of the RLS algorithm," *IEEE Trans. on Signal Processing*, vol. 39, pp. 583-594, March 1991.

5.9 PROBLEMS

1. The RLS algorithm is used to predict the signal $x(k) = \cos \frac{\pi k}{3}$ using a second-order FIR filter with the first tap fixed at 1. Given $\lambda = 0.98$, calculate the output signal $y(k)$ and the tap coefficients for the first 10 iterations. Note that we aim the minimization of $E[y^2(k)]$.

   Start with $\mathbf{w}^T(-1) = [1\ 0\ 0]$ and $\delta = 100$.

2. Show that the solution in equation (5.4) is a minimum point.

3. Show that $\mathbf{S}_D(k)$ approaches a null matrix for large k, when $\lambda = 1$.

4. Suppose that the measurement noise $n(k)$ is a random signal with zero-mean and the probability density with normal distribution. In a sufficient-order identification of an FIR system with optimal coefficients given by $\mathbf{w}_o$, show that the least-squares solution with $\lambda = 1$ is also normally distributed with mean $\mathbf{w}_o$ and covariance $E[\mathbf{S}_D(k)\sigma_n^2]$.

5. Prove that equation (5.42) is valid. What is the result when $n(k)$ has zero mean and is correlated to the input signal $x(k)$?

   Hint: You can use the relation $E[e^2(k)] = E[e(k)]^2 + \sigma^2[e(k)]$, where $\sigma^2[\cdot]$ means variance of $[\cdot]$.

6. Consider that the additive noise $n(k)$ is uncorrelated with the input and the desired signals and is also a nonwhite noise with autocorrelation matrix $\mathbf{R}_n$. Determine the transfer function of a prewhitening filter that applied to $d'(k) + n(k)$ and $x(k)$ generates the optimum least-squares solution $\mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p}$ for $k \to \infty$.

7. Show that if the additive noise is uncorrelated with $d'(k)$ and $x(k)$, and nonwhite, the least-squares algorithm will converge asymptotically to the optimal solution.

8. In problem 4, when $n(k)$ is correlated to $x(k)$, is $\mathbf{w}_o$ still the optimal solution? If not, what is the optimal solution?

9. Show that in the RLS algorithm the following relation is true

$$\xi^d(k) = \lambda\xi^d(k-1) + \varepsilon(k)e(k)$$

   where $e(k)$ is the *a priori* error as defined in equation (5.8).

10. Prove the validity of the approximation in equation (5.80).

11. Demonstrate that the updating formula for the complex RLS algorithm is given by equation (5.98).

12. Show that for an input signal with diagonal dominant correlation matrix $\mathbf{R}$ the following approximation related to equations (C.28) and (C.32) is valid.

$$E\{\mathbf{N}_{\mathbf{S}_D}(k)\mathbf{x}(k)\mathbf{x}^T(k)\mathrm{cov}[\Delta\mathbf{w}(k-1)_Q]\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{N}_{\mathbf{S}_D}(k)\} \approx \sigma^2_{\mathbf{S}_D}\sigma_x^4\mathrm{tr}\,\{\mathrm{cov}[\Delta\mathbf{w}(k-1)_Q]\}\,\mathbf{I}$$

13. Derive the equations (C.35), (C.36), and (C.37).

14. The conventional RLS 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 $\sigma^2_{\mathbf{w}} = 0.033$. The initial time-varying system multiplier coefficients are

$$\mathbf{w}_o^T = [0.03490\ -0.01100\ -0.06864\ 0.22391\ 0.55686\ 0.35798\ -0.02390\ -0.07594]$$

   The input signal is Gaussian white noise with variance $\sigma_x^2 = 1$ 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 $\lambda = 0.97$, compute the excess MSE.

(b) Repeat (a) for $\lambda = \lambda_{\text{opt}}$.

(c) Simulate the experiment described, measure the excess MSE, and compare to the calculated results.

15. Reduce the value of $\lambda_{\mathbf{w}}$ to 0.97 in problem 14, simulate, and comment on the results.

16. 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 conventional RLS algorithm. Consider that fixed-point arithmetic is used.

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	$\sigma_x^2 = 0.7$
	$\lambda = \lambda_{\text{opt}}$

$\mathbf{w}_o^T = [0.0219360\ \ 0.0015786\ \ -0.0602449\ \ -0.0118907\ \ 0.1375379$
$0.0574545\ \ -0.3216703\ \ -0.5287203\ \ -0.2957797\ \ 0.0002043\ \ 0.290670$
$-0.0353349\ \ -0.0068210\ \ 0.0026067\ \ 0.0010333\ \ -0.0143593]$

(a) Compute the expected value for $||\Delta\mathbf{w}(k)_Q||^2$ and $\xi(k)_Q$ for the described case.

(b) Simulate the identification example described and compare the simulated results with those obtained through the closed form formulas.

(c) Plot the learning curves for the finite- and infinite-precision implementations. Also, plot $E[||\Delta\mathbf{w}(k)||^2]$ versus k in both cases.

17. Repeat the above problem for the following cases

(a) $\sigma_n^2 = 0.01, b_c = 9$ bits, $b_d = 9$ bits, $\sigma_x^2 = 0.7, \lambda = \lambda_{\text{opt}}$.

(b) $\sigma_n^2 = 0.1, b_c = 10$ bits, $b_d = 10$ bits, $\sigma_x^2 = 0.8, \lambda = \lambda_{\text{opt}}$.

(c) $\sigma_n^2 = 0.05, b_c = 8$ bits, $b_d = 16$ bits, $\sigma_x^2 = 0.8, \lambda = \lambda_{\text{opt}}$.

18. In problem 17, compute (do not simulate) $E[||\Delta\mathbf{w}(k)_Q||^2], \xi(k)_Q$, and the probable number of iterations before the algorithm stop updating for $\lambda = 1, \lambda = 0.98, \lambda = 0.96$, and $\lambda = \lambda_{\text{opt}}$.

19. Repeat problem 16 for the case where the input signal is a first-order Markov process with $\lambda_{\mathbf{x}} = 0.95$.

20. A digital channel model can be represented by the following impulse response:

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

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 RLS algorithm. Use a filter of order 50 and plot the learning curve.

21. 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 equalizer. Again use the RLS algorithm.

22. Use the complex RLS algorithm to equalize a channel with the transfer function given below. The input signal is a four QAM 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 10 coefficients.

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

(a) Use an appropriate value for λ in the range $0.95 - 0.99$, run the algorithm and comment on the convergence behavior.
(b) Plot the real versus imaginary parts of the received signal before and after equalization.
(c) Increase the number of coefficients to 20 and repeat the experiment in (b).

23. 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

$$H(z) = 0.5 + 0.2\jmath + (-0.1 + 0.4\jmath)z^{-1} + (0.2 - 0.4\jmath)z^{-2} + (0.2 + 0.7\jmath)z^{-3}$$

The adaptive filter is also a third-order complex FIR filter, and the additional noise is zero-mean Gaussian white noise with variance $\sigma_n^2 = 0.3$. Using the complex RLS algorithm run an ensemble of 20 experiments, and plot the average learning curve.

24. Apply the Kalman filter to equalize the system

$$H(z) = \frac{0.19z}{z - 0.9}$$

when the additional noise is a uniformly distributed white noise with variance $\sigma_n^2 = 0.1$, and the input signal to the channel is a Gaussian noise with unit variance.