6

DATA-SELECTIVE ADAPTIVE FILTERING

6.1 INTRODUCTION

The families of adaptive-filtering algorithms introduced so far present a tradeoff between speed of convergence and the misadjustment after the transient. These characteristics are easily observable in stationary environments. In general fast converging algorithms tend to be very dynamic, a feature not necessarily advantageous after convergence in a stationary environment. In this chapter, an alternative formulation to govern the updating of the adaptive-filter coefficients is introduced. The basic assumption is that the additional noise is considered bounded, and the bound is either known or can be estimated [1]. The key strategy of the formulation is to find a *feasibility set*[1] such that the bounded error specification is met for any member of this set. As a result, the *set-membership filtering* (SMF) is aimed at estimating the feasibility set itself or a member of this set [2].

As a byproduct, the SMF allows the reduction of computational complexity in adaptive filtering, since the filter coefficients are updated only when the output estimation error is higher than the pre-determined upper bound [2]-[3].

Set-membership adaptive filters employ a deterministic objective function related to a bounded error constraint on the filter output, such that the updates belong to a set of feasible solutions. The objective function resembles the prescribed specifications of non-adaptive digital filter design. In the latter, any filter whose amplitude ripples in some frequency bands are smaller than given bounds is an acceptable solution. The main difference is that in the SMF the considered bound applies to the time-domain output error. As compared with their competing algorithms such as the normalized LMS, affine projection, and RLS algorithms [4]-[11], the SMF algorithms lead to reduced computational complexity primarily due to data-selective updates.

Usually the set-membership algorithms perform updates more frequently during the early iterations in stationary environments. As such, if these updates entail more computational complexity than available, some alternative solution is required. A possible strategy to maintain some control on the

[1]This set is defined as the set of filter coefficients leading to output errors whose moduli fall below a prescribed upper bound.

P.S.R. Diniz, *Adaptive Filtering*, DOI: 10.1007/978-0-387-68606-6_6,

amount of computational resources is to adopt partial update, where only a subset of the adaptive-filter coefficients are updated at each iteration. The resulting algorithms are collectively called partial-update (PU) algorithms [14]-[19].

This chapter presents several alternative set-membership algorithms which are closely related to the normalized LMS algorithm [7], the binormalized data-reusing LMS algorithm (here denoted as SM-BNLMS) [10], and the affine projection (SM-AP) [11] algorithm. In addition, this chapter describes the set-membership affine projection algorithm with partial update in some detail. The family of algorithms described in this chapter leads to more flexible management of the computational resources, in comparison with the algorithms presented in the previous chapters.

6.2 SET-MEMBERSHIP FILTERING

The SMF concept is a framework applicable to adaptive-filtering problems that are linear in parameters. The adaptive-filter output is given by

$$y(k) = \mathbf{w}^T \mathbf{x}(k) \tag{6.1}$$

where $\mathbf{x}(k) = [x_0(k)\ x_1(k)\ \ldots x_N(k)]^T$ is the input signal vector, and $\mathbf{w} = [w_0\ w_1\ \ldots w_N]^T$ is the parameter vector.

Considering a desired signal sequence $d(k)$ and a sequence of input vectors $\mathbf{x}(k)$, both for for $k = 0, 1, 2, \ldots, \infty$, the estimation error sequence $e(k)$ is calculated as

$$e(k) = d(k) - \mathbf{w}^T \mathbf{x}(k) \tag{6.2}$$

also for $k = 0, 1, 2, \ldots, \infty$. The vectors $\mathbf{x}(k)$ and $\mathbf{w} \in \mathbb{R}^{N+1}$, where $\mathbb{R}$ represents the set of real numbers, whereas $y(k)$ and $e(k)$ represent the adaptive-filter output signal and output error, respectively. The objective of the SMF is to design $\mathbf{w}$ such that the magnitude of estimation output error is upper bounded by a prescribed quantity $\bar{\gamma}$. If the value of $\bar{\gamma}$ is properly chosen there are several valid estimates for $\mathbf{w}$. In summary, any filter parameter leading to a magnitude of the output estimation error smaller than a deterministic threshold is an acceptable solution. From the bounded error constraint results a set of filters rather than a single estimate. If $\bar{\gamma}$ is chosen too small there might be no solution.

Assuming that $\bar{\mathcal{S}}$ denotes the set of all possible input-desired data pairs $(\mathbf{x}, d)$ of interest, it is possible to define Θ as the set of all possible vectors $\mathbf{w}$ leading to output errors whose magnitudes are bounded by $\bar{\gamma}$ whenever $(\mathbf{x}, d) \in \bar{\mathcal{S}}$. The set Θ, called *feasibility set*, is given by

$$\Theta = \bigcap_{(\mathbf{x},d)\in\bar{\mathcal{S}}} \{\mathbf{w} \in \mathbb{R}^{N+1} : |d - \mathbf{w}^T \mathbf{x}| \leq \bar{\gamma}\} \tag{6.3}$$

Now let's consider the practical case where only measured data are available. Given a set of data pairs $\{\mathbf{x}(i), d(i)\}$, for $i = 0, 1, \ldots, k$, let's define $\mathcal{H}(k)$ as the set containing all vectors $\mathbf{w}$ such that

the associated output error at time instant k is upper bounded in magnitude by $\bar{\gamma}$. That is,

$$\mathcal{H}(k) = \{\mathbf{w} \in \mathbb{R}^{N+1} : |d(k) - \mathbf{w}^T\mathbf{x}(k)| \leq \bar{\gamma}\} \tag{6.4}$$

The set $\mathcal{H}(k)$ is usually referred to as the *constraint set*. The boundaries of $\mathcal{H}(k)$ are hyperplanes. For the two-dimensional case, where the coefficient vector has two elements, $\mathcal{H}(k)$ comprises the region between the lines where $d(k) - \mathbf{w}^T\mathbf{x}(k) = \pm\bar{\gamma}$ as depicted in Fig. 6.1. For more dimensions, $\mathcal{H}(k)$ represents the region between two parallel hyperplanes in the parameter space $\mathbf{w}$.

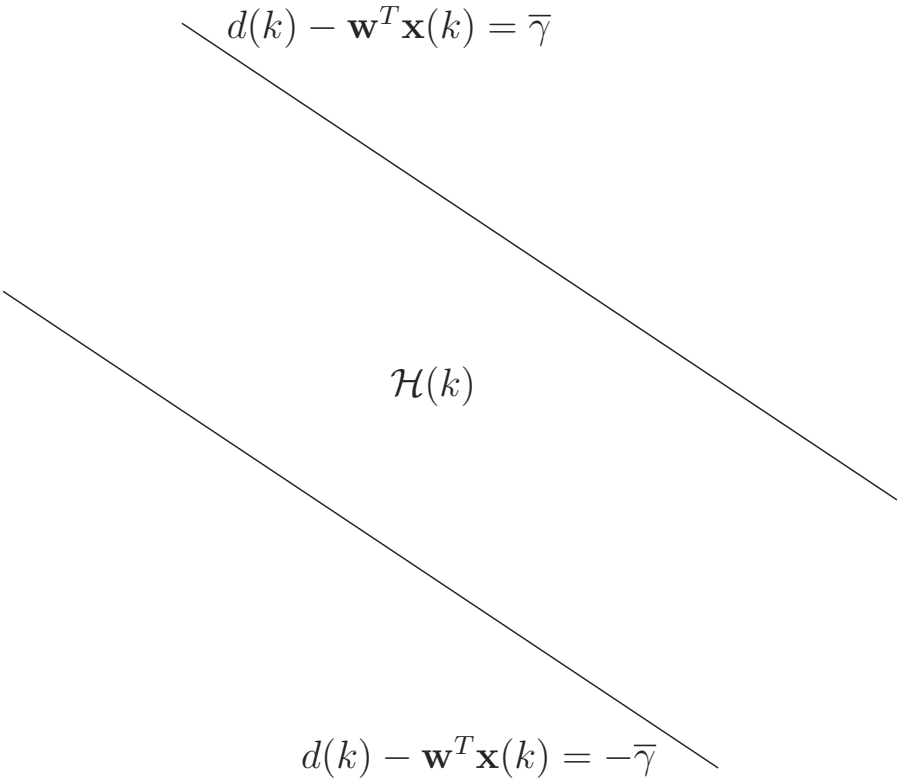


Figure 6.1 Constraint set in $\mathbf{w}$ plane for a two-dimension example.

Since for each data pair there is an associated constraint set, the intersection of the constraint sets over all the available time instants $i = 0, 1, \ldots, k$, is called the *exact membership set* $\psi(k)$, formally defined as

$$\psi(k) = \bigcap_{i=0}^{k} \mathcal{H}(i) \tag{6.5}$$

The set $\psi(k)$ represents a polygon in the parameter space whose location is one of the main objectives of the set-membership filtering.

For a set of data pairs including substantial innovation, the polygon in $\mathbf{w}$, $\psi(k)$, should become small. This property usually occurs after a large number of iterations k, when most likely $\psi(k) = \psi(k-1)$ since $\psi(k-1)$ is entirely contained in the constraint set $\mathcal{H}(k)$ as depicted in Fig. 6.2.a. In this case, the adaptive-filter coefficients do not need updating because the current membership set is totally inside the constraint set, resulting in a selection of update which is data dependent. The selective updating of the set-membership filtering brings about opportunities for power and computational savings, so crucial in devices such as mobile terminals. On the other hand, in the early iterations it is highly possible that the constraint set reduces the size of the membership-set polygon as illustrates Fig. 6.2.b.

At any given time instant, it can be observed that the feasibility set Θ is a subset of the exact membership set $\psi(k)$. The feasibility set is the *limiting set* of the exact membership set because the two sets are equal if the available input-desired data pairs traverses all signal pairs belonging to $\bar{\mathcal{S}}$.

The goal of set-membership adaptive filtering is to adaptively find an estimate that belongs to the feasibility set. The easiest approach is to compute a point estimate using, for example, the information provided by the constraint set $\mathcal{H}(k)$ like in the set-membership NLMS algorithm considered in the following section, or several previous constraint sets like the set-membership affine projection (SM-AP) algorithm discussed in section 6.4, [10], [11].

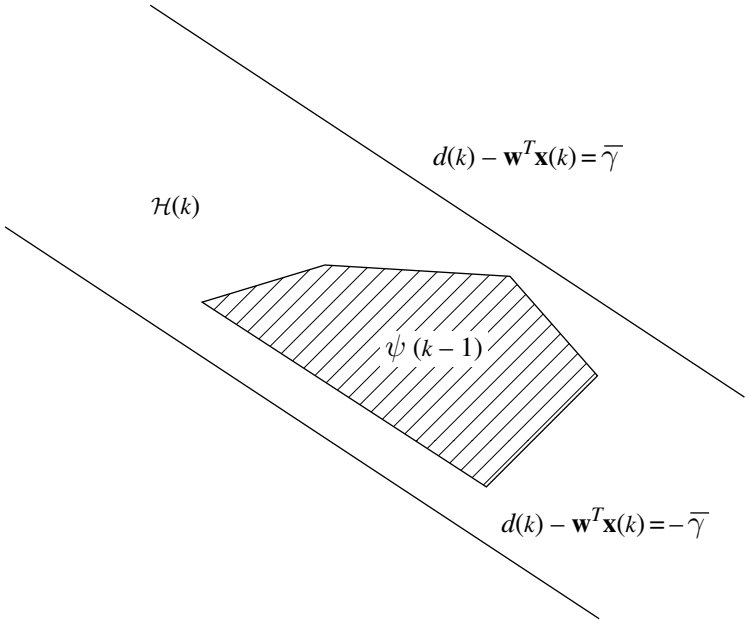
For historical reasons, it is worth mentioning that the first SMF approach proposed in the literature tries to outer bound $\psi(k)$ with ellipsoids and the resulting algorithms are called optimal bounding ellipsoid (OBE) algorithms [4]-[6]. These algorithms bear a close resemblance with the RLS algorithm [2] and have inherent data selectivity. In the OBE algorithms the membership set are bounded by ellipsoids comprising the smallest closed set [4]-[6]. These algorithms are also important but they are not included as they present higher computational complexity than those discussed here.

6.3 SET-MEMBERSHIP NORMALIZED LMS ALGORITHM

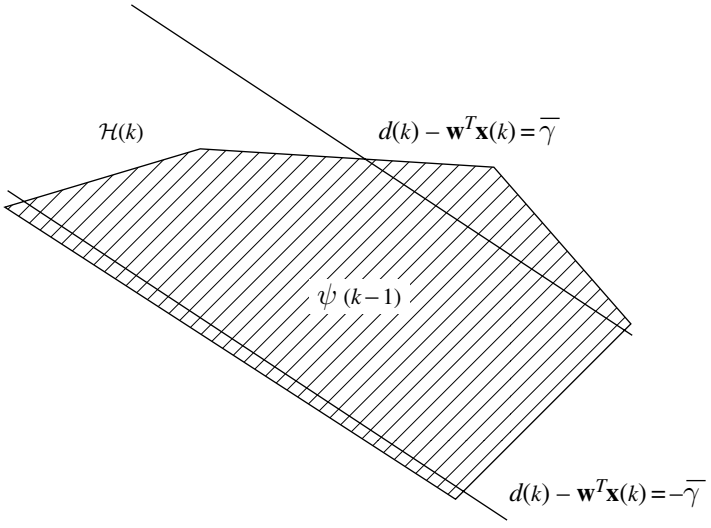
The set-membership NLMS (SM-NLMS) algorithm first proposed in [7] has a form similar to the conventional NLMS algorithm presented in section 4.4. The key idea of the SM-NLMS algorithm is to perform a test to verify if the previous estimate $\mathbf{w}(k)$ lies outside the constraint set $\mathcal{H}(k)$, i.e., $|d(k) - \mathbf{w}^T(k)\mathbf{x}(k)| > \bar{\gamma}$. If the modulus of the error signal is greater than the specified bound, the new estimate $\mathbf{w}(k+1)$ will be updated to the closest boundary of $\mathcal{H}(k)$ at a minimum distance, i.e., the SM-NLMS minimizes $\|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2$ subjected to $\mathbf{w}(k+1) \in \mathcal{H}(k)$ [12]. The updating is performed by an orthogonal projection of the previous estimate onto the closest boundary of $\mathcal{H}(k)$. Fig. 6.3 illustrates the updating procedure of the SM-NLMS algorithm.

In order to derive the update equations, first consider the *a priori* error $e(k)$ given by

$$e(k) = d(k) - \mathbf{w}^T(k)\mathbf{x}(k) \tag{6.6}$$



(a) Exact membership set, $\psi\ (k-1)$, contained in the constraint set, $\psi\ (k-1) \subset \mathcal{H}(k)$.



(b) Exact membership set, $\psi\ (k-1)$, not contained in the constraint set, $\psi\ (k-1) \not\subseteq \mathcal{H}(k)$.

Figure 6.2 Exact membership set $\psi(k)$ and its possible intersection with the constraint set $\mathcal{H}(k)$.

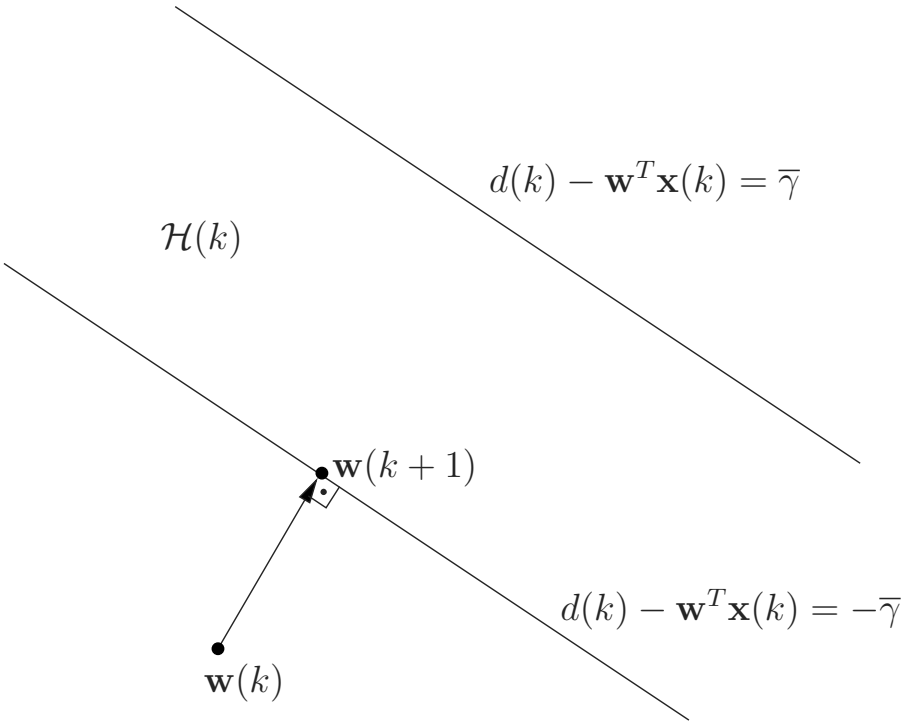


Figure 6.3 Coefficient vector updating for the set-membership normalized LMS algorithm.

then, let's start with the normalized LMS algorithm which utilizes the following recursion for updating $\mathbf{w}(k)$

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{\mu(k)}{\gamma + \mathbf{x}^T(k)\mathbf{x}(k)}\, e(k)\, \mathbf{x}(k) \tag{6.7}$$

where in the present discussion $\mu(k)$ is the variable step size that should be appropriately chosen in order to satisfy the desired set-membership updating.

The update should occur either if

$$e(k) = d(k) - \mathbf{w}^T(k)\mathbf{x}(k) > \bar{\gamma}$$

or

$$e(k) = d(k) - \mathbf{w}^T(k)\mathbf{x}(k) < -\bar{\gamma}$$

and the *a posteriori error* should be given by

$$\begin{aligned}
\varepsilon(k) &= d(k) - \mathbf{w}^T(k+1)\mathbf{x}(k) = \pm\bar{\gamma} \\
&= d(k) - \mathbf{w}^T(k)\mathbf{x}(k) - \frac{\mu(k)}{\gamma + \mathbf{x}^T(k)\mathbf{x}(k)}\, e(k)\, \mathbf{x}^T(k)\mathbf{x}(k) \\
&= e(k) - \frac{\mu(k)}{\gamma + \mathbf{x}^T(k)\mathbf{x}(k)}\, e(k)\, \mathbf{x}^T(k)\mathbf{x}(k)
\end{aligned} \tag{6.8}$$

where $\varepsilon(k)$ becomes equal to $\pm\bar{\gamma}$ because the coefficients are updated to the closest boundary of $\mathcal{H}(k)$. Since γ, whose only task is regularization, is a small constant it can be disregarded leading to the following equality

$$\varepsilon(k) = e(k)[1 - \mu(k)] = \pm\bar{\gamma} \tag{6.9}$$

The above equation leads to

$$1 - \mu(k) = \pm\frac{\bar{\gamma}}{e(k)} \tag{6.10}$$

where the plus $(+)$ sign applies for the case when $e(k) > 0$ and the minus $(-)$ sign applies for the case where $e(k) < 0$. Therefore, by inspection we conclude that the variable step size, $\mu(k)$, is given by

$$\mu(k) = \begin{cases} 1 - \frac{\bar{\gamma}}{|e(k)|} & \text{if } |e(k)| > \bar{\gamma} \\ 0 & \text{otherwise} \end{cases} \tag{6.11}$$

The updating equations (6.6), (6.11), and (6.7) are quite similar to those of the NLMS algorithm except for the variable step size $\mu(k)$. The SM-NLMS algorithm is outlined in Algorithm 6.1. As a rule of thumb, the value of $\bar{\gamma}$ is chosen around $\sqrt{5}\sigma_n$, where σ_n^2 is the variance of the additional noise, some further discussion in this matter is found in section 6.7 [7], [25].

The reader should recall that the NLMS algorithm minimizes $\|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2$ subjected to the constraint that $\mathbf{w}^T(k+1)\mathbf{x}(k) = d(k)$, as such it is a particular case of the SM-NLMS algorithm by choosing the bound $\bar{\gamma} = 0$. It should be noticed that by using a step size $\mu(k) = 1$ in the SM-NLMS whenever $\mathbf{w}(k) \notin \mathcal{H}(k)$, one performs a valid update since the hyperplane with zero *a posteriori* error lies in $\mathcal{H}(k)$. In this case, the resulting algorithm does not minimize the Euclidean distance $\|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2$ since the *a posteriori* error is zero and less than $\bar{\gamma}$.

6.4 SET-MEMBERSHIP AFFINE PROJECTION ALGORITHM

The exact membership set $\psi(k)$ previously defined in equation (6.5) suggests the use of more constraint-sets in the update [11]. This section generalizes the concept of set-membership in order to conceive algorithms whose updates belong to the past $L+1$ constraint sets. In order to achieve our goal, it is convenient to express $\psi(k)$ as

$$\psi(k) = \left(\bigcap_{i=0}^{k-L-1} \mathcal{H}(i) \right) \left(\bigcap_{j=k-L}^{k} \mathcal{H}(j) \right) = \psi^{k-L-1}(k) \bigcap \psi^{L+1}(k) \tag{6.12}$$

where $\psi^{L+1}(k)$ represents the intersection of the $L+1$ last constraint sets, whereas $\psi^{k-L-1}(k)$ is the intersection of the first $k-L$ constraint sets. The aim of this derivation is to conceive an algorithm whose coefficient update belongs to the last $L+1$ constraint-sets, i.e., $\mathbf{w}(k+1) \in \psi^{L+1}(k)$.

Just like in the original affine projection algorithm of section 4.6, we can retain the last $L+1$ input signal vectors in a matrix as follows:

$$\mathbf{X}_{\text{ap}}(k) = [\mathbf{x}(k)\, \mathbf{x}(k-1) \ldots \mathbf{x}(k-L)] \tag{6.13}$$

Algorithm 6.1

The Set-Membership Normalized LMS Algorithm

Initialization
$\mathbf{x}(0) = \mathbf{w}(0) = [0...0]^T$
choose $\bar{\gamma}$ around $\sqrt{5}\sigma_n$
$\gamma =$ small constant
Do for $k \geq 0$
$e(k) = d(k) - \mathbf{x}^T(k)\mathbf{w}(k)$
$\mu(k) = \begin{cases} 1 - \frac{\bar{\gamma}}{|e(k)|} & \text{if } |e(k)| > \bar{\gamma} \\ 0 & \text{otherwise} \end{cases}$
$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{\mu(k)}{\gamma + \mathbf{x}^T(k)\mathbf{x}(k)}\, e(k)\, \mathbf{x}(k)$

where $\mathbf{X}_{\text{ap}}(k) \in \mathbb{R}^{(N+1)\times(L+1)}$ contains the corresponding retained inputs, with $\mathbf{x}(k)$ being the input-signal vector

$$\mathbf{x}(k) = [x(k)\ x(k-1)\ \ldots\ x(k-N)]^T \tag{6.14}$$

The vectors representing the data considered at a given iteration k, such as the desired signal and error vectors are given by

$$\mathbf{d}_{\text{ap}}(k) = \begin{bmatrix} d(k) \\ d(k-1) \\ \vdots \\ d(k-L) \end{bmatrix} \tag{6.15}$$

$$\mathbf{e}_{\text{ap}}(k) = \begin{bmatrix} e_{\text{ap},0}(k) \\ e_{\text{ap},1}(k) \\ \vdots \\ e_{\text{ap},L}(k) \end{bmatrix} \tag{6.16}$$

where $\mathbf{d}_{\text{ap}}(k) \in \mathbb{R}^{(L+1)\times 1}$ contains the desired outputs from the $L+1$ last time instants.

Consider that $\mathcal{S}(k-i+1)$ denotes the hyperplane which contains all vectors $\mathbf{w}$ such that $d(k-i+1) - \mathbf{w}^T\mathbf{x}(k-i+1) = \bar{\gamma}_i(k)$, for $i = 1, \ldots, L+1$, where the parameters $\bar{\gamma}_i(k)$ represent the bound constraint to be satisfied by the error magnitudes after coefficient updating. Some particular choices for the parameters $\bar{\gamma}_i(k)$ are discussed later on, for now any choice satisfying the bound constraint is valid. That is, if all $\bar{\gamma}_i(k)$ are chosen such that $|\bar{\gamma}_i(k)| \leq \bar{\gamma}$ then $\mathcal{S}(k-i+1) \in \mathcal{H}(k-i+1)$, for $i = 1, \ldots, L+1$. Vector $\bar{\boldsymbol{\gamma}}(k) \in \mathbb{R}^{(L+1)\times 1}$ specifies the point in $\psi^{L+1}(k)$, where

$$\bar{\boldsymbol{\gamma}}(k) = [\bar{\gamma}_1(k)\ \bar{\gamma}_2(k)\ \ldots\ \bar{\gamma}_{L+1}(k)]^T \tag{6.17}$$

The objective function to be minimized in the set-membership affine projection (SM-AP) algorithm can now be stated. Perform a coefficient update whenever $\mathbf{w}(k) \notin \psi^{L+1}(k)$ in such a way that[2]

$$\min \|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2 \tag{6.18}$$

subject to:

$$\mathbf{d}_{\text{ap}}(k) - \mathbf{X}_{\text{ap}}^T(k)\mathbf{w}(k+1) = \bar{\boldsymbol{\gamma}}(k) \tag{6.19}$$

where the constraint can be rewritten as $d(k-i+1) - \mathbf{x}^T(k-i+1)\mathbf{w}(k+1) = \bar{\gamma}_i(k)$, for $i = 1, \ldots, L+1$. Fig. 6.4 illustrates a typical coefficient update related to the SM-AP algorithm for the case with two coefficients, $L = 1$ and $|\bar{\gamma}_i(k)| < |\bar{\gamma}|$, such that $\mathbf{w}(k+1)$ is not placed at the border of $\mathcal{H}(k)$.

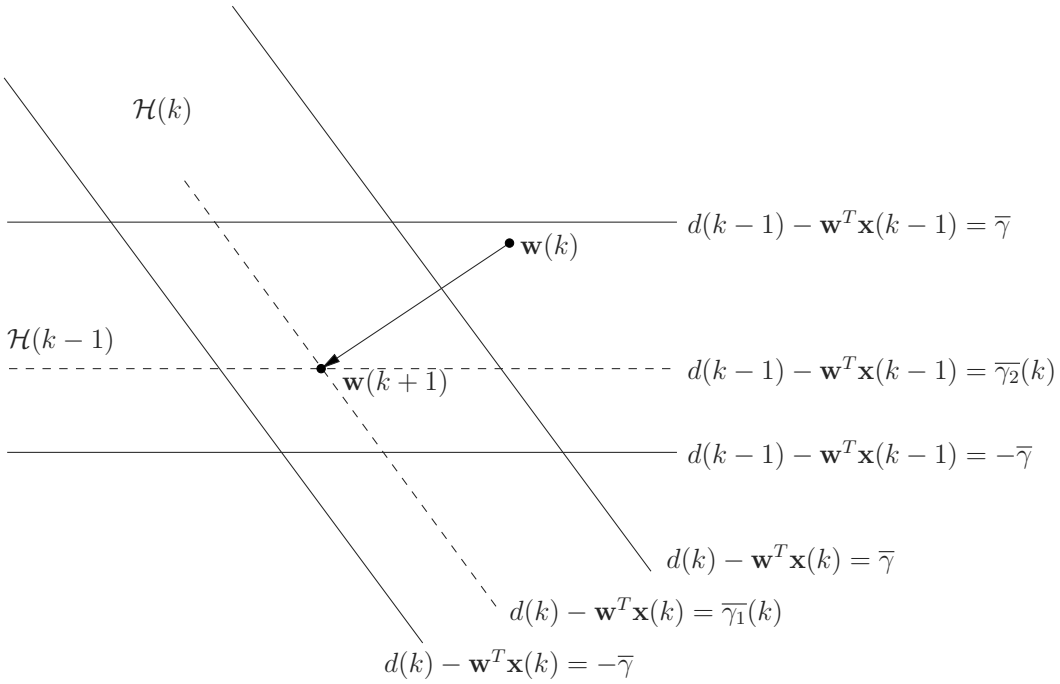


Figure 6.4 SM-AP algorithm coefficient update.

Using the method of Lagrange multipliers [12], the unconstrained function to be minimized is

$$F[\mathbf{w}(k+1)] = \|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2 + \boldsymbol{\lambda}_{\text{ap}}^T(k)[\mathbf{d}_{\text{ap}}(k) - \mathbf{X}_{\text{ap}}^T(k)\mathbf{w}(k+1) - \bar{\boldsymbol{\gamma}}(k)] \tag{6.20}$$

where the vector of Lagrange multipliers, $\boldsymbol{\lambda}_{\text{ap}}(k) \in \mathbb{R}^{(L+1)\times 1}$, is given by

$$\boldsymbol{\lambda}_{\text{ap}}(k) = [\lambda_{\text{ap},1}(k)\ \lambda_{\text{ap},2}(k) \ldots\ \lambda_{\text{ap},L+1}(k)]^T \tag{6.21}$$

[2]The reader should note that in earlier definition of the objective function related to the affine projection algorithm a constant $\frac{1}{2}$ was multiplied to the norm to be minimized. This constant is not relevant and is only used when it simplifies the algorithm derivation.

such that the constraints can be rewritten in the above equation as follows

$$\begin{aligned} F[\mathbf{w}(k+1)] &= \|\mathbf{w}(k+1)-\mathbf{w}(k)\|^2 \\ &\quad + \sum_{i=1}^{L+1} \lambda_{\mathrm{ap},i}(k)[d(k-i+1)-\mathbf{x}^T(k-i+1)\mathbf{w}(k+1)-\bar{\gamma}_i(k)] \end{aligned} \tag{6.22}$$

We solve the minimization problem of equation (6.18) by first setting the gradient of the function $F[\mathbf{w}(k+1)]$ with respect to $\mathbf{w}(k+1)$ equal to zero, in order to derive the following equation

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) + \sum_{i=1}^{L+1} \frac{\lambda_i(k)}{2}\mathbf{x}(k-i+1) \\ &= \mathbf{w}(k) + \mathbf{X}_{\mathrm{ap}}(k)\frac{\boldsymbol{\lambda}_{\mathrm{ap}}(k)}{2} \end{aligned} \tag{6.23}$$

By premultiplying the above equation by $\mathbf{X}_{\mathrm{ap}}^T(k)$ and utilizing the constraints

$$\mathbf{X}_{\mathrm{ap}}^T(k)\mathbf{w}(k+1) = \mathbf{d}_{\mathrm{ap}}(k) - \bar{\boldsymbol{\gamma}}(k)$$

given in equation (6.19), we obtain

$$\mathbf{d}_{\mathrm{ap}}(k) - \bar{\boldsymbol{\gamma}}(k) = \mathbf{X}_{\mathrm{ap}}^T(k)\mathbf{w}(k) + \mathbf{X}_{\mathrm{ap}}^T(k)\mathbf{X}_{\mathrm{ap}}(k)\frac{\boldsymbol{\lambda}_{\mathrm{ap}}(k)}{2} \tag{6.24}$$

or alternatively

$$d(k-i+1) - \bar{\gamma}_i(k) = \mathbf{x}^T(k-i+1)\mathbf{w}(k) + \sum_{j=1}^{L+1} \frac{\lambda_j(k)}{2}\mathbf{x}^T(k-i+1)\mathbf{x}(k-j+1)$$

for $i = 1, \ldots, L+1$.

Equation (6.24) can be rewritten in a more interesting format as

$$\begin{aligned} \mathbf{X}_{\mathrm{ap}}^T(k)\mathbf{X}_{\mathrm{ap}}(k)\frac{\boldsymbol{\lambda}_{\mathrm{ap}}(k)}{2} &= \mathbf{d}_{\mathrm{ap}}(k) - \mathbf{X}_{\mathrm{ap}}^T(k)\mathbf{w}(k) - \bar{\boldsymbol{\gamma}}(k) \\ &= \mathbf{e}_{\mathrm{ap}}(k) - \bar{\boldsymbol{\gamma}}(k) \end{aligned} \tag{6.25}$$

leading to

$$\frac{\boldsymbol{\lambda}_{\mathrm{ap}}(k)}{2} = \left[\mathbf{X}_{\mathrm{ap}}^T(k)\mathbf{X}_{\mathrm{ap}}(k)\right]^{-1}\left[\mathbf{e}_{\mathrm{ap}}(k) - \bar{\boldsymbol{\gamma}}(k)\right] \tag{6.26}$$

It is now possible to derive the updating equation by starting from equation (6.23) with $\boldsymbol{\lambda}_{\mathrm{ap}}(k)$ being given by equation (6.26), i.e.,

$$\mathbf{w}(k+1) = \begin{cases} \mathbf{w}(k) + \mathbf{X}_{\mathrm{ap}}(k)\left[\mathbf{X}_{\mathrm{ap}}^T(k)\mathbf{X}_{\mathrm{ap}}(k)\right]^{-1}\left[\mathbf{e}_{\mathrm{ap}}(k) - \bar{\boldsymbol{\gamma}}(k)\right] & \text{if } |e(k)| > \bar{\gamma} \\ \mathbf{w}(k) & \text{otherwise} \end{cases} \tag{6.27}$$

Algorithm 6.2

The Set-Membership Affine Projection Algorithm

Initialization

$\mathbf{x}(0) = \mathbf{w}(0) = [0 \ \ldots \ 0]^T$

choose $\bar{\gamma}$ around $\sqrt{5}\sigma_n$

$\gamma =$ small constant

Do for $k \geq 0$

$\mathbf{e}_{\mathrm{ap}}(k) = \mathbf{d}_{\mathrm{ap}}(k) - \mathbf{X}_{\mathrm{ap}}^T(k)\mathbf{w}(k)$

$$\mathbf{w}(k+1) = \begin{cases} \mathbf{w}(k) + \mathbf{X}_{\mathrm{ap}}(k)\left[\mathbf{X}_{\mathrm{ap}}^T(k)\mathbf{X}_{\mathrm{ap}}(k) + \gamma\mathbf{I}\right]^{-1}\left[\mathbf{e}_{\mathrm{ap}}(k) - \bar{\boldsymbol{\gamma}}(k)\right] & \text{if } |e(k)| > \bar{\gamma} \\ \mathbf{w}(k) & \text{otherwise} \end{cases}$$

where

$$\mathbf{e}_{\mathrm{ap}}(k) = [e(k) \ \varepsilon(k-1) \ \ldots \ \varepsilon(k-L)]^T \tag{6.28}$$

with $\varepsilon(k-i) = d(k-i) - \mathbf{x}^T(k-i)\mathbf{w}(k)$ denoting the *a posteriori* error calculated with the data pair of iteration $k-i$ using the coefficients of iteration k. Algorithm 6.2 describes in detail the general form of the SM-AP algorithm.

Several properties related to the SM-AP algorithm are straightforward to infer.

- For time instants $k < L+1$, i.e., during initialization, we can only assume knowledge of $\mathcal{H}(i)$ for $i = 0, 1, \ldots, k$. As a result, if an update is needed when $k < L+1$, the algorithm is used with the only $k+1$ constraint sets available.
- In order to verify if an update $\mathbf{w}(k+1)$ is required, we only have to check if $\mathbf{w}(k) \notin \mathcal{H}(k)$ since due to previous updates $\mathbf{w}(k) \in \mathcal{H}(k-i+1)$ holds for $i = 2, \ldots, L+1$.
- By choosing the bound $\bar{\gamma} = 0$, it is possible to verify that the algorithm becomes the conventional AP algorithm with unity step-size.

6.4.1 A Trivial Choice for Vector $\bar{\boldsymbol{\gamma}}(k)$

In the above discussions no specific choice for the parameters $\bar{\gamma}_i(k)$ has been discussed except for the requirement that the adaptive-filter coefficients should be in $\mathcal{H}(k-i+1)$, meaning that $|\bar{\gamma}_i(k)| \leq \bar{\gamma}$. There is infinite number of possible choices for $\bar{\gamma}_i(k)$, each leading to a different update.

The most trivial choice would be $\bar{\boldsymbol{\gamma}}(k) = \mathbf{0}$, i.e., to force the *a posteriori* errors to be zero at the last $L+1$ time instants. If we replace $\bar{\boldsymbol{\gamma}}(k) = \mathbf{0}$ in equation (6.24) and solving for $\boldsymbol{\lambda}_{\mathrm{ap}}(k)$ the following recursions result

$$\frac{\boldsymbol{\lambda}_{\mathrm{ap}}(k)}{2} = \left(\mathbf{X}_{\mathrm{ap}}^T(k)\mathbf{X}_{\mathrm{ap}}(k)\right)^{-1}\mathbf{e}_{\mathrm{ap}}(k) \tag{6.29}$$

The update recursion is given by

$$\mathbf{w}(k+1) = \begin{cases} \mathbf{w}(k) + \mathbf{X}_{\mathrm{ap}}(k)\left(\mathbf{X}_{\mathrm{ap}}^T(k)\mathbf{X}_{\mathrm{ap}}(k)\right)^{-1}\mathbf{e}_{\mathrm{ap}}(k) & \text{if } |e(k)| > \bar{\gamma} \\ \mathbf{w}(k) & \text{otherwise} \end{cases} \tag{6.30}$$

The above updating equation is identical to the conventional affine-projection (AP) algorithm with unity step size whenever an update takes place, that is, $\mathbf{w}(k) \notin \mathcal{H}(k)$. However, owing to the data selectivity, the SM-AP algorithm leads to considerable reduction in complexity as compared with the conventional AP algorithm. Fig. 6.5 depicts a typical coefficient update, where for illustration purposes $\mathbf{w}(k)$ does not lie in the zero *a posteriori* hyperplane belonging to $\mathcal{H}(k-1)$.

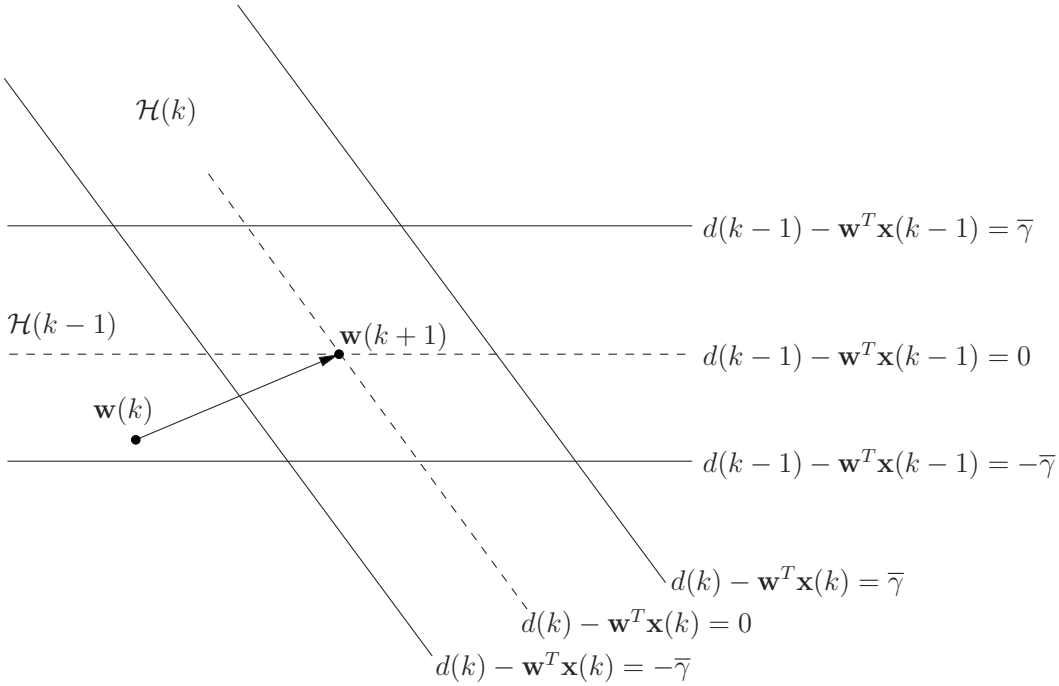


Figure 6.5 SM-AP algorithm coefficient updated with zero *a posteriori* error.

6.4.2 A Simple Vector $\bar{\boldsymbol{\gamma}}(k)$

Any choice for $\bar{\gamma}_i(k)$ is valid as long as they correspond to points represented by the adaptive-filter coefficients in $\mathcal{H}(k-i+1)$, i.e., $|\bar{\gamma}_i(k)| \leq \bar{\gamma}$. One can exploit this freedom in order to make the

resulting algorithm more suitable for a target application. A particularly simple SM-AP version is obtained if $\bar{\gamma}_i(k)$ for $i \neq 1$ corresponds to the *a posteriori* error $\varepsilon(k-i+1) = d(k-i+1) - \mathbf{w}^T(k)\mathbf{x}(k-i+1)$ and $\bar{\gamma}_1(k) = e(k)/|e(k)|$. Since the coefficients were updated considering previous data pairs then at this point it is true that $\mathbf{w}(k) \in \mathcal{H}(k-i+1)$, i.e., $|\varepsilon(k-i+1)| = |d(k-i+1) - \mathbf{x}^T(k-i+1)\mathbf{w}(k)| \leq \bar{\gamma}$, for $i = 2, \ldots, L+1$. Therefore, by choosing $\bar{\gamma}_i(k) = \varepsilon(k-i+1)$, for $i \neq 1$, all the elements on the right-hand side of equation (6.24) become zero, except for first element.

It is only left now the choice of the constraint value $\bar{\gamma}_1(k)$, that can be selected as in the SM-NLMS algorithm where $\bar{\gamma}_1(k)$ is such that the solution lies at the nearest boundary of $\mathcal{H}(k)$, i.e.,

$$\bar{\gamma}_1(k) = \bar{\gamma}\frac{e(k)}{|e(k)|} \tag{6.31}$$

Such choices utilized in equation (6.25) leads to

$$\mathbf{X}_{\text{ap}}^T(k)\mathbf{X}_{\text{ap}}(k)\frac{\boldsymbol{\lambda}_{\text{ap}}(k)}{2} = \mu(k)e(k)\mathbf{u}_1 \tag{6.32}$$

where $\mu(k) = 1 - \frac{\bar{\gamma}}{|e(k)|}$ and $\mathbf{u}_1 = [1\ 0\ \ldots\ 0]^T$.

The resulting update equation is then given by

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{X}_{\text{ap}}(k)\left[\mathbf{X}_{\text{ap}}^T(k)\mathbf{X}_{\text{ap}}(k)\right]^{-1}\mu(k)e(k)\mathbf{u}_1 \tag{6.33}$$

where

$$e(k) = d(k) - \mathbf{w}^T(k)\mathbf{x}(k) \tag{6.34}$$

$$\mu(k) = \begin{cases} 1 - \frac{\bar{\gamma}}{|e(k)|} & \text{if } |e(k)| > \bar{\gamma} \\ 0 & \text{otherwise} \end{cases} \tag{6.35}$$

This algorithm minimizes the Euclidean distance $\|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2$ subject to the constraint $\mathbf{w}(k+1) \in \psi^{L+1}(k)$ such that the *a posteriori* errors at iteration $k-i$, $\varepsilon(k-i)$, are kept constant for $i = 2, \ldots, L+1$. Fig. 6.6 illustrates a typical coefficient updating for the simplified SM-AP algorithm where it is observed that the *a posteriori* error related to previous data remains unaltered.

The simplified SM-AP algorithm given by equation (6.33) will perform an update if and only if $\mathbf{w}(k) \notin \mathcal{H}(k)$, or $e(k) > \bar{\gamma}$. The step-by-step description of the simplified SM-AP algorithm is presented in Algorithm 6.3.

6.4.3 Reducing the Complexity in the Simplified SM-AP Algorithm

In the updating expression of equation (6.33) vector $\mathbf{u}_1$ has a special form which can be exploited in order to reduce the computational complexity.

The inverse matrix in equation (6.33) can be partitioned as

$$\begin{aligned}\left[\mathbf{X}_{\text{ap}}^T(k)\mathbf{X}_{\text{ap}}(k)\right]^{-1} &= \left\{\left[\mathbf{x}(k)\ \tilde{\mathbf{X}}_{\text{ap}}(k)\right]^T\left[\mathbf{x}(k)\ \tilde{\mathbf{X}}_{\text{ap}}(k)\right]\right\}^{-1} \\ &= \begin{bmatrix} a & \mathbf{b}^T \\ \mathbf{b} & \mathbf{C} \end{bmatrix}\end{aligned} \tag{6.36}$$

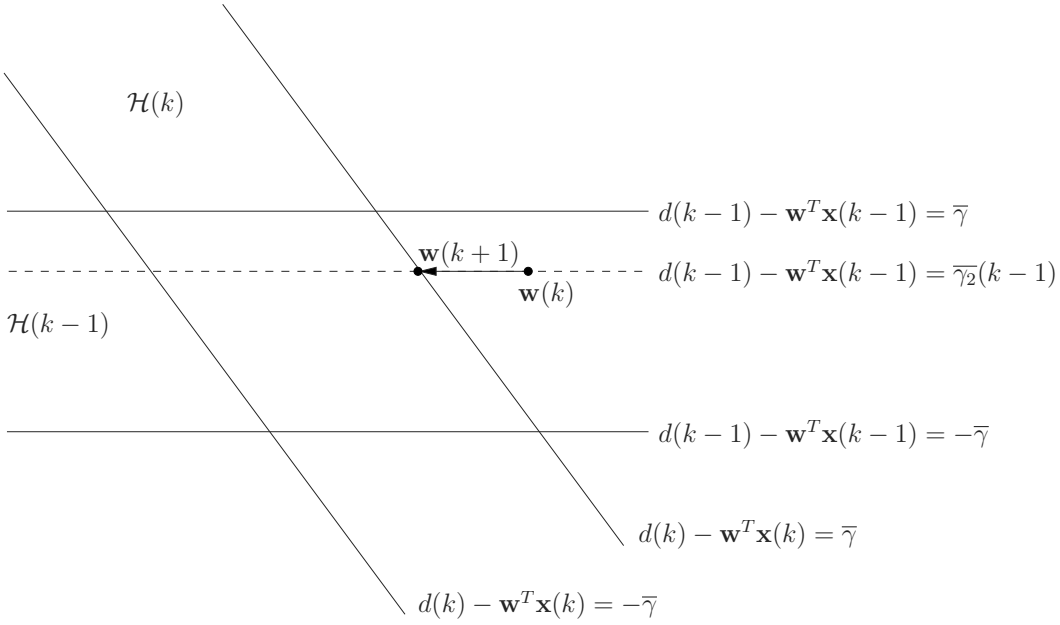


Figure 6.6 Simplified SM-AP algorithm coefficient update with constant *a posteriori* error.

Algorithm 6.3

The Simplified Set-Membership Affine Projection Algorithm

Initialization

$\mathbf{x}(0) = \mathbf{w}(0) = [0 \ \ldots \ 0]^T$

choose $\bar{\gamma}$ around $\sqrt{5}\sigma_n$

$\gamma =$ small constant

Do for $k \geq 0$

$\mathbf{e}_{\mathrm{ap}}(k) = \mathbf{d}_{\mathrm{ap}}(k) - \mathbf{X}_{\mathrm{ap}}^T(k)\mathbf{w}(k)$

$$\mu(k) = \begin{cases} 1 - \frac{\bar{\gamma}}{|e(k)|} & \text{if } |e(k)| > \bar{\gamma} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{X}_{\mathrm{ap}}(k)\left[\mathbf{X}_{\mathrm{ap}}^T(k)\mathbf{X}_{\mathrm{ap}}(k) + \gamma\mathbf{I}\right]^{-1}\mu(k)e(k)\mathbf{u}_1$$

where

$$a = \left[\boldsymbol{\varphi}^T(k)\boldsymbol{\varphi}(k)\right]^{-1} \tag{6.37}$$

$$\mathbf{b} = -\left[\tilde{\mathbf{X}}_{\mathrm{ap}}^T(k)\tilde{\mathbf{X}}_{\mathrm{ap}}(k)\right]^{-1}\tilde{\mathbf{X}}_{\mathrm{ap}}^T(k)\mathbf{x}(k)a \tag{6.38}$$

with $\boldsymbol{\varphi}(k)$ defined as

$$\boldsymbol{\varphi}(k) = \mathbf{x}(k) - \tilde{\mathbf{X}}_{\mathrm{ap}}(k)\left[\tilde{\mathbf{X}}_{\mathrm{ap}}^T(k)\tilde{\mathbf{X}}_{\mathrm{ap}}(k)\right]^{-1}\tilde{\mathbf{X}}_{\mathrm{ap}}^T(k)\mathbf{x}(k) \tag{6.39}$$

where the vector $\boldsymbol{\varphi}(k) \in \mathbb{R}^{(N+1)\times 1}$, see problem 12.

As a result,

$$\begin{aligned}
\mathbf{X}_{\mathrm{ap}}(k)\left[\mathbf{X}_{\mathrm{ap}}^T(k)\mathbf{X}_{\mathrm{ap}}(k)\right]^{-1}\mathbf{u}_1 &= \left[\mathbf{x}(k)\ \tilde{\mathbf{X}}_{\mathrm{ap}}(k)\right]\begin{bmatrix} a \\ \mathbf{b}\end{bmatrix} \\
&= \left[\mathbf{x}(k)\ \tilde{\mathbf{X}}_{\mathrm{ap}}(k)\right]\begin{bmatrix} 1 \\ \frac{\mathbf{b}}{a}\end{bmatrix} a \\
&= \left[\mathbf{x}(k) - \left[\tilde{\mathbf{X}}_{\mathrm{ap}}^T(k)\tilde{\mathbf{X}}_{\mathrm{ap}}(k)\right]^{-1}\tilde{\mathbf{X}}_{\mathrm{ap}}^T(k)\mathbf{x}(k)\right] a \\
&= \boldsymbol{\varphi}(k)\left[\boldsymbol{\varphi}^T(k)\boldsymbol{\varphi}(k)\right]^{-1}
\end{aligned} \tag{6.40}$$

where the last equality follows from equations (6.37) and (6.39).

An efficient expression for the coefficient update is obtained using the partition in equation (6.36), that is

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{\boldsymbol{\varphi}(k)}{\boldsymbol{\varphi}^T(k)\boldsymbol{\varphi}(k)}e(k) \tag{6.41}$$

where $\boldsymbol{\varphi}(k)$ is defined as in equation (6.39). This representation of the SM-AP algorithm is computationally attractive since it utilizes matrices with lower dimensions than those presented in equation (6.33), specifically matrix $\left[\tilde{\mathbf{X}}_{\mathrm{ap}}^T(k)\tilde{\mathbf{X}}_{\mathrm{ap}}(k)\right]$ in equation (6.39) has dimension $L \times L$ whereas matrix $\left[\mathbf{X}_{\mathrm{ap}}^T(k)\mathbf{X}_{\mathrm{ap}}(k)\right]$ in equation (6.33) has dimension $(L+1)\times(L+1)$. The number of reuses L is in most of the cases chosen in the range $0 \leq L \leq 5$, therefore the strategy for reducing the computational burden of the inversion brings about significant benefit.

6.5 SET-MEMBERSHIP BINORMALIZED LMS ALGORITHMS

In the SM-AP algorithm the computational complexity is directly related to the number of data reuses. The main component of the computational burden is the information matrix inversion. Since the SM-NLMS algorithm only considers the constraint set $\mathcal{H}(k)$ in its update, it has low complexity per update whereas its convergence speed follows the pattern of the NLMS algorithm. Both algorithms have

their convergence speed governed by the eigenvalue spread of the input-signal correlation matrix. In order to alleviate this drawback while keeping the implementation complexity as low as possible, an attractive particular solution for the SM-AP algorithm is the set-membership binormalized LMS (SM-BNLMS) algorithm. Two algorithms are derived requiring that the solution belongs to the constraint sets at time instants k and $k-1$, i.e., $\mathbf{w}(k+1) \in \mathcal{H}(k) \cap \mathcal{H}(k-1)$, which are general cases of the binormalized LMS algorithm [13]. The SM-BNLMS algorithms can be seen as extensions of the SM-NLMS algorithm that use two consecutive constraint sets for each update, and also as special cases of the SM-AP algorithms.

Let's assume $\mathcal{S}(k-i+1)$, for $i = 1, 2$, denote the hyperplanes which contain all vectors $\mathbf{w}$ such that $d(k-i+1) - \mathbf{w}^T\mathbf{x}(k-i+1) = \bar{\gamma}_i(k)$, where $\bar{\gamma}_i(k)$ are the values of the bound constraints that should be met in order to validate a given estimate. Specifically, if $\bar{\gamma}_i(k)$, for $i = 1, 2$, are chosen such that $|\bar{\gamma}_i(k)| \leq \bar{\gamma}$, then $\mathcal{S}(k-i+1) \in \mathcal{H}(k-i+1)$ [10].

Whenever $\mathbf{w}(k) \notin \mathcal{H}(k) \cap \mathcal{H}(k-1)$, we can propose an objective function such as

$$\begin{aligned} &\min \|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2 \\ &\text{subject to:} \\ &d(k) - \mathbf{x}^T(k)\mathbf{w}(k+1) = \bar{\gamma}_1(k) \\ &d(k-1) - \mathbf{x}^T(k-1)\mathbf{w}(k+1) = \bar{\gamma}_2(k) \end{aligned} \tag{6.42}$$

where the pair of thresholds $(\bar{\gamma}_1(k), \bar{\gamma}_2(k))$ specifies the point in $\mathcal{H}(k) \cap \mathcal{H}(k-1)$ where the final parameter estimate will be placed. The previously shown Fig. 6.4 illustrates how the coefficients are updated to prescribed *a posteriori* errors determined by $(\bar{\gamma}_1(k), \bar{\gamma}_2(k))$.

In principle there is a need to verify if an update according to equation (6.42) is required, where such an update can be skip if $\mathbf{w}(k) \in \mathcal{H}(k) \cap \mathcal{H}(k-1)$. There are ways of keeping $\mathbf{w}(k+1) \in \mathcal{H}(k-1)$ whenever an update is required, that is, whenever $\mathbf{w}(k) \notin \mathcal{H}(k)$. This type of solution is discussed further in subsection 6.5.2. At any rate, we can solve the general constrained minimization problem of equation (6.42) for the binormalized case by applying Lagrange multiplier method, resulting in the following unconstrained objective function

$$\begin{aligned} F[\mathbf{w}(k+1)] \;=\; &\|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2 + \lambda_1(k)[d(k) - \mathbf{x}^T(k)\mathbf{w}(k+1) - \bar{\gamma}_1(k)] \\ &+\lambda_2(k)[d(k-1) - \mathbf{x}^T(k-1)\mathbf{w}(k+1) - \bar{\gamma}_2(k)] \end{aligned} \tag{6.43}$$

By computing the gradient of equation (6.43) with respect to $\mathbf{w}(k+1)$, setting the result to zero, we get

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) + \mathbf{X}_{\text{ap}}(k)\frac{\boldsymbol{\lambda}_{\text{ap}}(k)}{2} \\ &= \mathbf{w}(k) + [\mathbf{x}(k)\,\mathbf{x}(k-1)] \begin{bmatrix} \frac{\lambda_1(k)}{2} \\ \frac{\lambda_2(k)}{2} \end{bmatrix} \end{aligned} \tag{6.44}$$

where this expression is the specialized form of equation (6.23) to the binormalized case.

The Lagrange multipliers are obtained by replacing equation (6.44) in the constraints of equation (6.42) such that

$$\begin{bmatrix} \mathbf{x}^T(k) \\ \mathbf{x}^T(k-1) \end{bmatrix} [\mathbf{x}(k)\, \mathbf{x}(k-1)] \begin{bmatrix} \frac{\lambda_1(k)}{2} \\ \frac{\lambda_2(k)}{2} \end{bmatrix} = \begin{bmatrix} d(k) \\ d(k-1) \end{bmatrix} - \begin{bmatrix} \mathbf{x}^T(k) \\ \mathbf{x}^T(k-1) \end{bmatrix} \mathbf{w}(k) - \bar{\boldsymbol{\gamma}}(k)$$
$$= \begin{bmatrix} e(k) \\ \varepsilon(k-1) \end{bmatrix} - \begin{bmatrix} \bar{\gamma}_1(k) \\ \bar{\gamma}_2(k) \end{bmatrix} \tag{6.45}$$

By solving the above equation we obtain

$$\frac{\lambda_1(k)}{2} = \frac{[e(k) - \bar{\gamma}_1(k)]\, \|\mathbf{x}(k-1)\|^2 - [\varepsilon(k-1) - \bar{\gamma}_2(k)]\, \mathbf{x}^T(k)\mathbf{x}(k-1)}{\|\mathbf{x}(k)\|^2 \|\mathbf{x}(k-1)\|^2 - [\mathbf{x}^T(k-1)\mathbf{x}(k)]^2} \tag{6.46}$$

$$\frac{\lambda_2(k)}{2} = \frac{[\varepsilon(k-1) - \bar{\gamma}_2(k)]\, \|\mathbf{x}(k)\|^2 - [e(k) - \bar{\gamma}_1(k)]\, \mathbf{x}^T(k-1)\mathbf{x}(k)}{\|\mathbf{x}(k)\|^2 \|\mathbf{x}(k-1)\|^2 - [\mathbf{x}^T(k-1)\mathbf{x}(k)]^2} \tag{6.47}$$

where the errors in the above equations are the *a priori* error at iteration k, defined as $e(k) = d(k) - \mathbf{w}^T(k)\mathbf{x}(k)$, and the *a posteriori* error at iteration $k-1$, defined as $\varepsilon(k-1) = d(k-1) - \mathbf{w}^T(k)\mathbf{x}(k-1)$.

The expression for the coefficient update of the SM-BNLMS algorithm is then given by

$$\mathbf{w}(k+1) = \begin{cases} \mathbf{w}(k) + \frac{\lambda_1(k)}{2}\mathbf{x}(k) + \frac{\lambda_2(k)}{2}\mathbf{x}(k-1) & \text{if } |e(k)| > \bar{\gamma} \\ \mathbf{w}(k) & \text{otherwise} \end{cases} \tag{6.48}$$

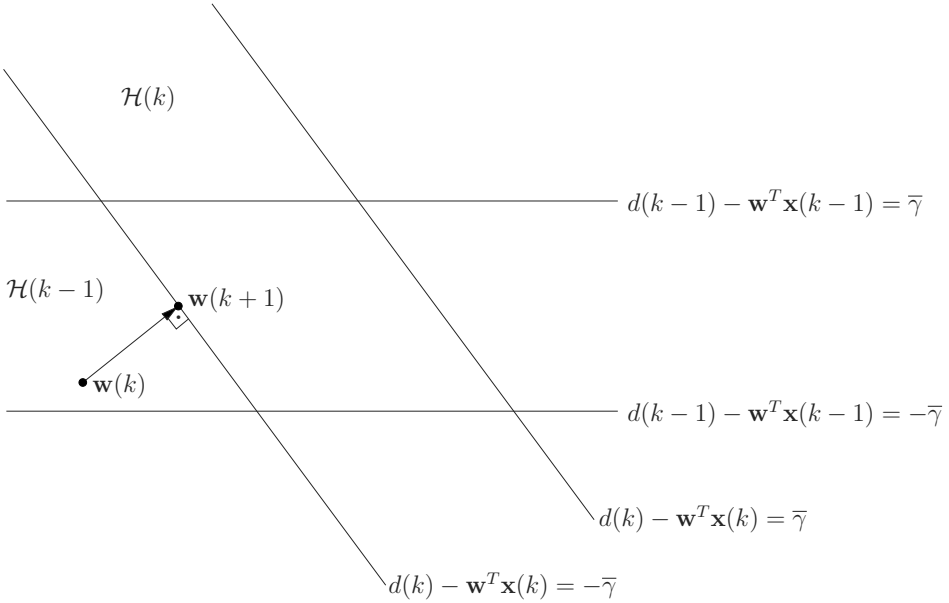
Some special forms of the SM-BNLMS algorithm are following discussed.

6.5.1 SM-BNLMS Algorithm 1

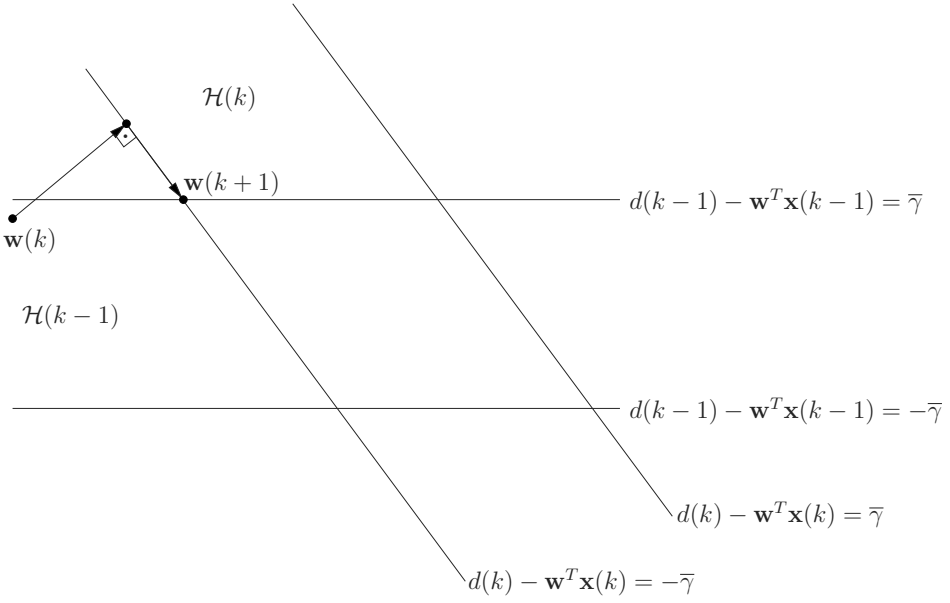
The first form of the SM-BNLMS algorithm is derived by employing two steps, where in each step we minimize the Euclidean distance between the old filter coefficients and the new update, subjected to the constraint that the new update lies in constraint set $\mathcal{H}(k)$. Then, we test if the new update belongs in the previous constraint set $\mathcal{H}(k-1)$ and if not a new update takes place. Basically, the SM-BNLMS algorithm 1 performs a step according to the SM-NLMS algorithm and if the solution belongs to both constraint sets $\mathcal{H}(k)$ and $\mathcal{H}(k-1)$ no further update is required. If the initial step moves the solution away from $\mathcal{H}(k-1)$, then a second update is performed in order to place the solution at the intersection of $\mathcal{H}(k)$ and $\mathcal{H}(k-1)$ at a minimum distance from $\mathbf{w}(k)$. Fig. 6.7 illustrates the coefficient updates according to the situations discussed so far. As desired, the SM-BNLMS algorithm 1 minimizes $\|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2$ subject to the constraint that $\mathbf{w}(k+1) \in \mathcal{H}(k) \cap \mathcal{H}(k-1)$.

The updating equation for the SM-BNLMS algorithm 1 can be derived by first performing an orthogonal projection of $\mathbf{w}(k)$ onto the nearest boundary of $\mathcal{H}(k)$ just like in the SM-NLMS algorithm

$$\hat{\mathbf{w}}(k) = \mathbf{w}(k) + \mu(k)\frac{e(k)\mathbf{x}(k)}{\|\mathbf{x}(k)\|^2} \tag{6.49}$$



(a) $\mathbf{w}(k+1) \in \mathcal{H}(k) \cap \mathcal{H}(k-1)$, no further update.



(b) $\mathbf{w}(k+1) \notin \mathcal{H}(k-1)$, further update.

Figure 6.7 Possible coefficient updates for the SM-BNLMS algorithm 1.

where $\mu(k)$ is the variable convergence factor given by equation (6.11) and $e(k)$ is the *a priori* output error defined in equation (6.6). If $\hat{\mathbf{w}}(k) \in \mathcal{H}(k-1)$, i.e., $|d(k-1) - \hat{\mathbf{w}}^T(k)\mathbf{x}(k-1)| \leq \bar{\gamma}$, no further update is required, therefore $\mathbf{w}(k+1) = \hat{\mathbf{w}}(k)$. On the other hand, if $\hat{\mathbf{w}}(k) \notin \mathcal{H}(k-1)$ a second step is necessary in order to move the solution to the intersection of $\mathcal{H}(k)$ and $\mathcal{H}(k-1)$ at a minimum distance. This second step is performed in the orthogonal direction with respect to the first step, namely $\mathbf{x}^{\perp}(k)$. The resulting second updating is then performed in the following form

$$\mathbf{w}(k+1) = \hat{\mathbf{w}}(k) + \hat{\mu}(k)\frac{\varepsilon(k-1)\mathbf{x}^{\perp}(k)}{\|\mathbf{x}^{\perp}(k)\|^2} \tag{6.50}$$

where

$$\mathbf{x}^{\perp}(k) = \left(\mathbf{I} - \frac{\mathbf{x}(k)\mathbf{x}^T(k)}{\|\mathbf{x}(k)\|^2}\right)\mathbf{x}(k-1) \tag{6.51}$$

$$\varepsilon(k-1) = d(k-1) - \hat{\mathbf{w}}^T(k)\mathbf{x}(k-1) \tag{6.52}$$

$$\hat{\mu}(k) = 1 - \frac{\bar{\gamma}}{|\varepsilon(k-1)|} \tag{6.53}$$

Algorithm 6.4 describes in detail the SM-BNLMS algorithm 1, where we utilized an explicit form for $\mathbf{x}^{\perp}(k)$, see problem 2. It is straightforward to observe that if the bound of the estimation error is chosen to be zero, i.e., $\bar{\gamma} = 0$, the updating equations of the SM-BNLMS algorithm 1 coincide with those of binormalized LMS algorithm with unity step-size [13].

In the SM-BNLMS algorithm 1 if the constraint sets $\mathcal{H}(k)$ and $\mathcal{H}(k-1)$ are parallel, the denominator term $\|\mathbf{x}^{\perp}(k)\|^2$ is zero, since this term is given by

$$\|\mathbf{x}^{\perp}(k)\|^2 = \|\mathbf{x}(k-1)\|^2 - \frac{[\mathbf{x}^T(k-1)\mathbf{x}(k)]^2}{\|\mathbf{x}(k)\|^2}$$

As a result the second step of equation (6.50) is not performed to avoid division by zero.

6.5.2 SM-BNLMS Algorithm 2

The SM-BNLMS algorithm 2 reduces the computational complexity per update even further by avoiding the intermediate constraint check required by the SM-BNLMS algorithm 1. A smart idea to avoid extra computation is, at instant k, to maintain the value of the *a posteriori* error $\varepsilon(k-1)$, which utilizes the data from instant $k-1$, equal to the constraint threshold, that is by choosing $\bar{\gamma}_2(k) = \varepsilon(k-1)$. Since the previous coefficient estimate $\mathbf{w}(k) \in \mathcal{H}(k-1)$, then it is a fact that $\varepsilon(k-1) \leq \bar{\gamma}$. Therefore by choosing $\bar{\gamma}_2(k) = \varepsilon(k-1)$ then $\bar{\gamma}_2(k) \leq \bar{\gamma}$. On the other hand, if we choose $\bar{\gamma}_1(k)$ such that the update lies on the closest boundary of $\mathcal{H}(k)$, i.e., $\bar{\gamma}_1(k) = \bar{\gamma}\text{sign}[e(k)]$, the new coefficient estimate $\mathbf{w}(k+1)$ lies on the nearest boundary of $\mathcal{H}(k)$ such that the *a posteriori* error at iteration $k-1$, $\varepsilon(k-1)$, is kept constant. By specializing the updating equation of the general SM-BNLMS algorithm to the SM-BNLMS algorithm 2 case, we have

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{\lambda_1'(k)}{2}\mathbf{x}(k) + \frac{\lambda_2'(k)}{2}\mathbf{x}(k-1) \tag{6.54}$$

Algorithm 6.4

The Set-Membership Binormalized LMS Algorithm 1

Initialization

$\mathbf{x}(0) = \mathbf{w}(0) = [0...0]^T$

choose $\bar{\gamma}$ around $\sqrt{5}\sigma_n$

$\gamma =$ small constant

Do for $k \geq 0$

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

$\mu(k) = \begin{cases} 1 - \frac{\bar{\gamma}}{|e(k)|} & \text{if } |e(k)| > \bar{\gamma} \\ 0 & \text{otherwise} \end{cases}$

$\hat{\mathbf{w}}(k) = \mathbf{w}(k) + \mu(k)\frac{e(k)\mathbf{x}(k)}{\gamma + \|\mathbf{x}(k)\|^2}$

$\varepsilon(k-1) = d(k-1) - \hat{\mathbf{w}}^T(k)\mathbf{x}(k-1)$

$\hat{\mu}(k) = \begin{cases} 1 - \frac{\bar{\gamma}}{|\varepsilon(k-1)|}, & \text{if } |e(k)| > \bar{\gamma} \text{ and } |\varepsilon(k-1)| > \bar{\gamma} \\ 0 & \text{otherwise} \end{cases}$

$\frac{\lambda_1(k)}{2} = -\frac{\hat{\mu}(k)\varepsilon(k-1)\mathbf{x}^T(k-1)\mathbf{x}(k)}{\gamma + \|\mathbf{x}(k)\|^2\|\mathbf{x}(k-1)\|^2 - [\mathbf{x}^T(k-1)\mathbf{x}(k)]^2}$

$\frac{\lambda_2(k)}{2} = \frac{\hat{\mu}(k)\varepsilon(k-1)\|\mathbf{x}(k)\|^2}{\gamma + \|\mathbf{x}(k)\|^2\|\mathbf{x}(k-1)\|^2 - [\mathbf{x}^T(k-1)\mathbf{x}(k)]^2}$

$\mathbf{w}(k+1) = \hat{\mathbf{w}}(k) + \frac{\lambda_1(k)}{2}\mathbf{x}(k) + \frac{\lambda_2(k)}{2}\mathbf{x}(k-1)$

where

$$\frac{\lambda_1'(k)}{2} = \frac{\mu(k)e(k)\|\mathbf{x}(k-1)\|^2}{\|\mathbf{x}(k)\|^2\|\mathbf{x}(k-1)\|^2 - [\mathbf{x}^T(k-1)\mathbf{x}(k)]^2} \tag{6.55}$$

$$\frac{\lambda_2'(k)}{2} = -\frac{\mu(k)e(k)\mathbf{x}^T(k-1)\mathbf{x}(k)}{\|\mathbf{x}(k)\|^2\|\mathbf{x}(k-1)\|^2 - [\mathbf{x}^T(k-1)\mathbf{x}(k)]^2} \tag{6.56}$$

$$\mu(k) = \begin{cases} 1 - \frac{\bar{\gamma}}{|e(k)|}, & \text{if } |e(k)| > \bar{\gamma} \\ 0 & \text{otherwise} \end{cases} \tag{6.57}$$

Fig. 6.6 depicts the update procedure of the SM-BNLMS algorithm 2, whereas Algorithm 6.5 describes it stepwise.

In the SM-BNLMS algorithm 2 if the constraint sets $\mathcal{H}(k)$ and $\mathcal{H}(k-1)$ are parallel, the denominators of the $\lambda_i'(k)$, for $i = 1, 2$ are zero. In this case, in order to avoid division by zero a regularization factor, as in equation (6.7), is employed instead.

Algorithm 6.5

The Set-Membership Binormalized LMS Algorithm 2

Initialization

$\mathbf{x}(0) = \mathbf{w}(0) = [0...0]^T$

choose $\bar{\gamma}$ around $\sqrt{5}\sigma_n$

$\gamma =$ small constant

Do for $k \geq 0$

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

$$\mu(k) = \begin{cases} 1 - \frac{\bar{\gamma}}{|e(k)|} & \text{if } |e(k)| > \bar{\gamma} \\ 0 & \text{otherwise} \end{cases}$$

$$\frac{\lambda_1'(k)}{2} = \frac{\mu(k)e(k)\|\mathbf{X}(k-1)\|^2}{\gamma + \|\mathbf{X}(k)\|^2\|\mathbf{X}(k-1)\|^2 - [\mathbf{X}^T(k-1)\mathbf{X}(k)]^2}$$

$$\frac{\lambda_2'(k)}{2} = -\frac{\mu(k)e(k)\mathbf{X}^T(k-1)\mathbf{X}(k)}{\gamma + \|\mathbf{X}(k)\|^2\|\mathbf{X}(k-1)\|^2 - [\mathbf{X}^T(k-1)\mathbf{X}(k)]^2}$$

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{\lambda_1'(k)}{2}\mathbf{x}(k) + \frac{\lambda_2'(k)}{2}\mathbf{x}(k-1)$$

6.6 COMPUTATIONAL COMPLEXITY

A brief comparison of the computational complexity among some algorithms presented in this chapter is appropriate at this point. The figure of merit considered is the number of multiplications, additions, and divisions, where it is assumed that the implementation minimizes the number of divisions, multiplications, and additions in that order. Table 6.1 lists the computational complexity for several algorithms, where in the case of the SM-BNLMS algorithm 1 there are two entries since the update complexity is related to the number of steps a given update requires. Two steps are required if after the first step $\hat{\mathbf{w}}(k) \notin \mathcal{H}(k-1)$. The SM-BNLMS algorithm 2 has fix complexity whenever an update occurs whereas for the SM-BNLMS algorithm 1 the complexity depends not only on when an update occurs but also how often the second step takes place. As expected the two versions of the SM-BNLMS algorithm lead to a small increase in computational complexity when compared with the SM-NLMS algorithm. On the other hand, the former algorithms usually require less updates and converge faster than the SM-NLMS algorithm.

The computational complexity reduction is essential in applications where the filter order is high and the resources are limited. Therefore, special care should be taken to exploit opportunities to reduce the computational burden, for example, assuming the value of $\|\mathbf{x}(k-1)\|^2$ at iteration k is unknown. If $\|\mathbf{x}(k-1)\|^2$ is known, we can compute $\|\mathbf{x}(k)\|^2$ using only two additional multiplications through $\|\mathbf{x}(k)\|^2 = \|\mathbf{x}(k-1)\|^2 + x^2(k) - x^2(k-N)$, also in case the value of $x^2(k-N)$ is prestored then only one multiplication is required. This strategy has been considered when evaluating the multiplication and addition counts of the SM-BNLMS algorithms. If update occurs at two successive time instants, $\|\mathbf{x}(k-1)\|^2$ and $\mathbf{x}^T(k-1)\mathbf{x}(k-2)$ have already been computed in the previous update, as a result,

Table 6.1 Computational Complexity in Set-membership Algorithms, † The Numbers for the RLS Apply to the Particular Implementation of Algorithm 5.2

Algorithm	Multiplication	Addition	Division
LMS	$2N+3$	$2N+2$	0
NLMS	$2N+3$	$2N+5$	1
SM-NLMS	$2N+4$	$2N+6$	1
SM-BNLMS 1 (1 step)	$3N+4$	$3N+7$	1
SM-BNLMS 1 (2 steps)	$5N+13$	$5N+16$	2
SM-BNLMS 2	$3N+11$	$3N+10$	1
RLS†	$3N^2+11N+8$	$3N^2+7N+4$	1

the number of multiplications and additions in such updates can be further reduced by approximately $N+1$ for the SM-NLMS algorithm and $2N+2$ for the SM-BNLMS algorithms 1 and 2, depending on the implementation. Finally, note that if one continuously computes $\|\mathbf{x}(k)\|^2$ and $\mathbf{x}^T(k)\mathbf{x}(k-1)$, regardless if an update is required or not, the SM-BNLMS algorithm 2 is always more efficient than SM-BNLMS algorithm 1.

6.7 TIME-VARYING $\bar{\gamma}$

In this section, an automatic way to choose $\bar{\gamma}$ is presented in order to avoid overbounding and underbounding of such a crucial parameter. In case $\bar{\gamma}$ is chosen too small the feasibility set might become null, whereas if the threshold parameter is chosen too big the resulting estimate might be meaningless and inconsistent [24].

Let's first consider the case of channel equalization application such as that of Fig. 2.13. In a typical multiuser communication environment the noise signal vector can be composed as follows [32]

$$\mathbf{n}(k) = \mathbf{n}_{\mathrm{n}}(k) + \mathbf{n}_{\mathrm{ISI}}(k) + \mathbf{n}_{\mathrm{MAI}}(k) \tag{6.58}$$

where $\mathbf{n}(k) = [n(k) \;\; n(k-1) \;\; \ldots \;\; n(k-N)]^T$, and

- $\mathbf{n}_{\mathrm{n}}(k)$ represents the contribution of the environment noise.
- $\mathbf{n}_{\mathrm{ISI}}(k)$ is the contribution of the intersymbol interference (ISI) originated when the transmitted signal crosses a channel with memory, in other words, whenever multiple paths are perceived by the receiver.
- $\mathbf{n}_{\mathrm{MAI}}(k)$ accounts for the multi-access interference (MAI), that is, the signals from other users that reach the receiver.

At the equalizer output, the disturbance due to noise can be accounted for as follows

$$y_{\mathrm{n}}(k) = \mathbf{w}^T(k)\mathbf{n}(k) \tag{6.59}$$

where $\mathbf{w}^T(k)$ is the equalizer coefficient vector and $y_{\mathrm{n}}(k)$ is the noise signal vector filtered by the equalizer. As a result, the equalizer output $y(k)$ is described by

$$y(k) = y_{\bar{\mathrm{n}}}(k) + y_{\mathrm{n}}(k) \tag{6.60}$$

with $y_{\bar{\mathrm{n}}}(k)$ representing the equalized signal when there is no noise at the adaptive-filter input.

The average power of the disturbance, for a given equalizer with parameters $\mathbf{w}(k)$, can be calculated as

$$\sigma^2_{y_{\mathrm{n}}}(k) = E[y^2_{\mathrm{n}}(k)] = \mathbf{w}^T(k)E[\mathbf{n}(k)\mathbf{n}^T(k)]\mathbf{w}(k) = \|\mathbf{w}(k)\|^2\sigma^2_{\mathrm{n}}(k) \tag{6.61}$$

Assuming there is an estimate of $\sigma^2_{y_{\mathrm{n}}}(k)$ denoted as $\hat{\sigma}^2_{y_{\mathrm{n}}}(k) = \|\mathbf{w}(k)\|^2\hat{\sigma}^2_{\mathrm{n}}(k)$ we can generate a time-varying threshold parameter as follows

$$\bar{\gamma}(k+1) = \alpha\bar{\gamma}(k) + (1-\alpha)\sqrt{\beta\|\mathbf{w}(k)\|^2\hat{\sigma}^2_{\mathrm{n}}(k)} \tag{6.62}$$

where α is a forgetting factor and β is a constant to be set. As justified in [25], a range of values for β leading to a good compromise between misadjustment and speed of convergence is $4 \leq \beta \leq 5$.

In equalization environments the best way to estimate $\sigma^2_{\mathrm{n}}(k)$ is to remove the effect of the detected symbols from $x(k)$ in order to get a rough estimate of $n(k)$ [26]-[28], and from this estimate compute

$$\hat{\sigma}^2_{\mathrm{n}}(k+1) = \alpha\hat{\sigma}^2_{\mathrm{n}}(k) + (1-\alpha)\hat{n}^2(k) \tag{6.63}$$

where again α is a forgetting factor. Fig. 6.8 illustrates how the environment noise can be typically estimated in a general equalizer setup.

For system identification environment as depicted in Fig. 2.10, an estimate of the additional noise plus an eventual effect of undermodeling can be calculated from the output error itself. If the input signal and the additional noise are considered white noise and uncorrelated, see equation (2.148) for details, the MSE can be calculated as

$$\begin{aligned}\xi &= E[e^2(k)] \\ &= E\{[\mathbf{h}^T\mathbf{x}_\infty(k) - \mathbf{w}^T\mathbf{x}_{N+1}(k)]^2 + n^2(k)\} \\ &= \sigma_x^2 \sum_{i=N+1}^{\infty} h^2(i) + \sigma_n^2\end{aligned} \tag{6.64}$$

where $\mathbf{x}_\infty(k)$ and $\mathbf{x}_{N+1}(k)$ are the input signal vector with infinite and finite lengths, respectively. Likewise the equalization setup, a time-varying threshold parameter for the system identification application is given by

$$\bar{\gamma}(k+1) = \alpha\bar{\gamma}(k) + (1-\alpha)\sqrt{\beta\hat{\sigma}^2_{\mathrm{n}}(k)} \tag{6.65}$$

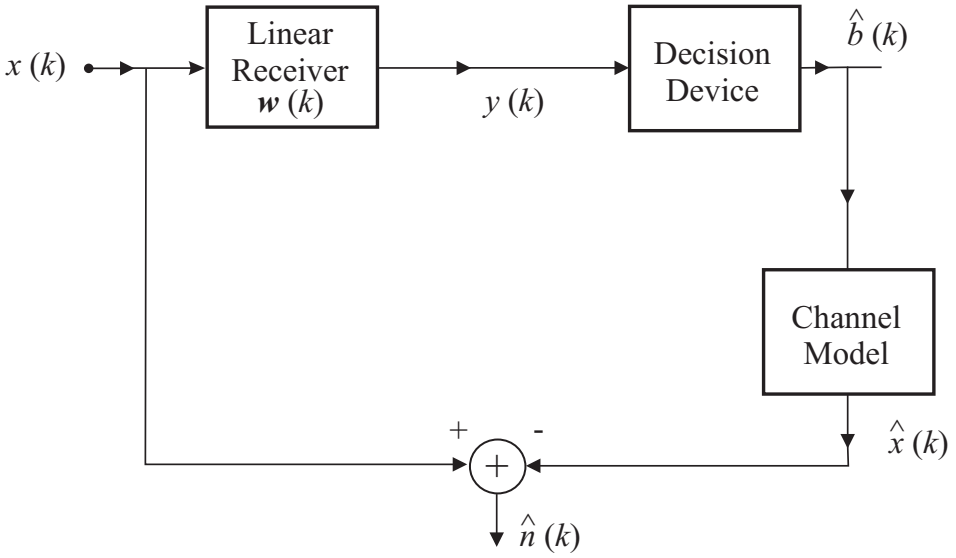


Figure 6.8 Environment noise estimation.

where for this case

$$\hat{\sigma}_{\mathrm{n}}^2(k+1) = \alpha\hat{\sigma}_{\mathrm{n}}^2(k) + (1-\alpha)e^2(k) \tag{6.66}$$

In [25] some analytical expressions are developed in order to provide values for $\bar{\gamma}(k)$ such that the some prescribed updating rate are nearly satisfied after the algorithm has reached convergence.

6.8 PARTIAL-UPDATE ADAPTIVE FILTERING

In several applications the number of coefficients to be updated might be prohibitive, therefore some strategies to control the computational complexity is desirable. In some cases like in acoustics echo cancellation, which might use a few thousands of adaptive coefficients, the convergence would entail a large number of iterations, calling for more sophisticated updating algorithms which are inherently more computationally intensive. A good compromise might be to update only part of the filter coefficients at each iteration instant, generating a family of algorithms called partial-update (PU) algorithms. The most widely known PU algorithm in the literature is the normalized LMS with partial update [14]-[19].

In this section special emphasis is given to the set-membership partial-update affine projection (SM-PUAP) algorithms. The combination of the partial-update with set-membership allows the updating of a selected set of coefficients whenever an update is needed. The resulting algorithms capitalize not only from the sparse updating related to the set-membership framework but also from the partial

update of the coefficients, reducing the average computational complexity. It is expected that the SM-PUAP algorithms have comparable performance to that of SM-AP algorithms and affine projection algorithms with partial-update whereas computational complexity is reduced with respect to both updating schemes.

Two versions of the SM-PUAP algorithm are discussed:

- Fix partial update, where a constant number of coefficients is updated whenever required.
- Variable partial update, where the number of coefficients to be updated vary up to a maximum prescribed number.

In the partial update adaptation strategy, the main objective is to perform updates in $\bar{M}$ out of the $N+1$ adaptive-filter coefficients. The $\bar{M}$ coefficients to be updated at time instant k are selected through an index set $\mathcal{I}_{\bar{M}}(k) = \{i_0(k) \ \ldots \ i_{\bar{M}-1}(k)\}$ where the indexes $\{i_j(k)\}_{j=0}^{\bar{M}-1}$ are chosen from the set $\{0 \ 1 \ \ldots \ N\}$ representing the available coefficients to be updated. The partition of the $N+1$ coefficients into mutually exclusive subsets, each with $\bar{M}$ elements, plays a key role in the performance and in the effectiveness of the partial-update strategy. As a result, $\mathcal{I}_{\bar{M}}(k)$ varies with the iteration index k such that the $\bar{M}$ coefficients to be updated can change according to the iteration. The choice of which $\bar{M}$ coefficients should be updated is related to the objective function considered in the algorithm derivation.

As already known, in the SM-AP algorithms the new coefficient vector can be obtained as the vector $\mathbf{w}(k+1)$ that minimizes the Euclidean distance $\|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2$, subject to the constraint that the moduli of *a posteriori* errors fall below certain prescribed threshold. The same idea can be used in order to derive the SM-PUAP algorithm, specifically the vector $\mathbf{w}(k+1)$ is chosen by minimizing the Euclidean distance $\|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2$ subject to the constraint $\mathbf{w}(k+1) \in \mathcal{H}(k)$ in such a way that only $\bar{M}$ coefficients are updated. If $\mathbf{w}(k) \in \mathcal{H}(k)$, there is no update and the Euclidean distance is zero.

The objective function to be minimized in the set-membership partial-update affine projection (SM-PUAP) algorithm is following described. A coefficient update is performed whenever $\mathbf{w}(k) \notin \psi^{L+1}(k)$ such that

$$\min \|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2 \tag{6.67}$$

subject to:

$$\mathbf{d}_{\mathrm{ap}}(k) - \mathbf{X}_{\mathrm{ap}}^T(k)\mathbf{w}(k+1) = \bar{\boldsymbol{\gamma}}(k)$$

$$\tilde{\mathbf{C}}_{\mathcal{I}_{\bar{M}(k)}} [\mathbf{w}(k+1) - \mathbf{w}(k)] = \mathbf{0} \tag{6.68}$$

where $\bar{\boldsymbol{\gamma}}(k)$ is a vector determining a point within the constraint set $\mathcal{H}(k)$, such that $|\bar{\gamma}_i(k)| \leq \bar{\gamma}$, for $i = 0, 1, \ldots, L$. The matrix $\tilde{\mathbf{C}}_{\mathcal{I}_{\bar{M}(k)}} = \mathbf{I} - \mathbf{C}_{\mathcal{I}_{\bar{M}}(k)}$ is a complementary matrix of $\mathbf{C}_{\mathcal{I}_{\bar{M}}(k)}$ enforcing $\tilde{\mathbf{C}}_{\mathcal{I}_{\bar{M}}(k)}\mathbf{w}(k+1) = \tilde{\mathbf{C}}_{\mathcal{I}_{\bar{M}}(k)}\mathbf{w}(k)$, such that only $\bar{M}$ coefficients are updated. A possible choice for $\bar{\gamma}_0(k)$ is such that the updated vector belongs to the closest bounding hyperplane in $\mathcal{H}(k)$, i.e., $\bar{\gamma}_0(k) = \bar{\gamma}\, e(k)/|e(k)|$. On the other hand, some alternative choices $|\bar{\gamma}_i(k)| \leq \bar{\gamma}$, for $i = 1, 2, \ldots, L$, had been discussed. The matrix $\mathbf{C}_{\mathcal{I}_{\bar{M}}(k)}$ is a diagonal matrix that determines the coefficients to be

updated at instant k, if an update is required. This matrix has $\bar{M}$ nonzero elements equal to one placed at positions indicated by $\mathcal{I}_{\bar{M}}(k)$.

Applying the method of Lagrange multipliers gives the recursive updating rule

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{C}_{\mathcal{I}_{\bar{M}}(k)}\mathbf{X}_{\mathrm{ap}}(k)\left[\mathbf{X}_{\mathrm{ap}}^T(k)\mathbf{C}_{\mathcal{I}_{\bar{M}}(k)}\mathbf{X}_{\mathrm{ap}}(k)\right]^{-1}\left[\mathbf{e}_{\mathrm{ap}}(k) - \bar{\boldsymbol{\gamma}}(k)\right] \tag{6.69}$$

The updating equation of the SM-PUAP algorithm is given by

$$\mathbf{w}(k+1) = \begin{cases} \mathbf{w}(k) + \mathbf{C}_{\mathcal{I}_{\bar{M}}(k)}\mathbf{X}_{\mathrm{ap}}(k)\left[\mathbf{X}_{\mathrm{ap}}^T(k)\mathbf{C}_{\mathcal{I}_{\bar{M}}(k)}\mathbf{X}_{\mathrm{ap}}(k)\right]^{-1}\left[\mathbf{e}_{\mathrm{ap}}(k) - \bar{\boldsymbol{\gamma}}(k)\right] & \text{if } |e(k)| > \bar{\gamma} \\ \mathbf{w}(k) & \text{otherwise} \end{cases} \tag{6.70}$$

As can be noticed from equation (6.70), for a fixed value of $\|\mathbf{e}_{\mathrm{ap}}(k) - \bar{\boldsymbol{\gamma}}(k)\|^2$, the Euclidean distance between two consecutive coefficient vectors is minimized if $\|\mathbf{X}_{\mathrm{ap}}^T(k)\mathbf{C}_{\mathcal{I}_{\bar{M}}(k)}\mathbf{X}_{\mathrm{ap}}(k)\|$ is maximized. As a result, a natural choice for the $\bar{M}$ coefficients to be updated are those that will be multiplied by the elements of $\mathbf{X}_{\mathrm{ap}}(k)$ with the largest norm.

Like in the case of the SM-AP algorithm of equation (6.33), it is straightforward to derive a simplified version of the SM-PUAP algorithm, whose update equation is given by

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{C}_{\mathcal{I}_{\bar{M}}(k)}\mathbf{X}_{\mathrm{ap}}(k)\left[\mathbf{X}_{\mathrm{ap}}^T(k)\mathbf{C}_{\mathcal{I}_{\bar{M}}(k)}\mathbf{X}_{\mathrm{ap}}(k)\right]^{-1}\mu(k)e(k)\mathbf{u}_1 \tag{6.71}$$

where

$$e(k) = d(k) - \mathbf{w}^T(k)\mathbf{x}(k) \tag{6.72}$$

$$\mu(k) = \begin{cases} 1 - \frac{\bar{\gamma}}{|e(k)|} & \text{if } |e(k)| > \bar{\gamma} \\ 0 & \text{otherwise} \end{cases} \tag{6.73}$$

This algorithm also minimizes the Euclidean distance $\|\mathbf{w}(k+1) - \mathbf{w}(k)\|^2$ subject to the constraint $\mathbf{w}(k+1) \in \psi^{L+1}(k)$ maintaining the values of the *a posteriori* errors, $\varepsilon(k-i)$, at iteration $k-i$. Note that $\mu(k)$ starts with high values, becomes small when the error reduces, and reaches zero whenever moduli of the errors become smaller than the threshold. An interesting choice for the index set $\mathcal{I}_{\bar{M}}(k)$ specifying the coefficients to be updated is the $\bar{M}$ coefficients leading to the maximum value of $\|\mathbf{X}_{\mathrm{ap}}^T(k)\mathbf{C}_{\mathcal{I}_{\bar{M}}(k)}\mathbf{X}_{\mathrm{ap}}(k)\|$. Algorithm 6.6 describes in detail the simplified version of the SM-PUAP algorithm.

6.8.1 Set-Membership Partial-Update NLMS Algorithm

The simplest form of the SM-PUAP algorithm is the set-membership partial-update NLMS (SM-PUNLMS) algorithm. The updating equation of the SM-PUNLMS algorithm follows directly from

Algorithm 6.6

The Simplified Set-Membership Partial-Update Affine Projection Algorithm

Initialization

$\mathbf{x}(0) = \mathbf{w}(0) = [0 \ \ldots \ 0]^T$

choose $\bar{\gamma}$ around $\sqrt{5}\sigma_n$

$\gamma =$ small constant

Do for $k \geq 0$

$\mathbf{e}_{\mathrm{ap}}(k) = \mathbf{d}_{\mathrm{ap}}(k) - \mathbf{X}_{\mathrm{ap}}^T(k)\mathbf{w}(k)$

$\mu(k) = \begin{cases} 1 - \frac{\bar{\gamma}}{|e(k)|} & \text{if } |e(k)| > \bar{\gamma} \\ 0 & \text{otherwise} \end{cases}$

$\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{C}_{\mathcal{I}_{\bar{M}}(k)}\mathbf{X}_{\mathrm{ap}}(k)\left[\mathbf{X}_{\mathrm{ap}}^T(k)\mathbf{C}_{\mathcal{I}_{\bar{M}}(k)}\mathbf{X}_{\mathrm{ap}}(k) + \gamma\mathbf{I}\right]^{-1}\mu(k)e(k)\mathbf{u}_1$

equation (6.71) and is given by

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu(k)\frac{e(k)\mathbf{C}_{\mathcal{I}_{\bar{M}}(k)}\mathbf{x}(k)}{\|\mathbf{C}_{\mathcal{I}_{\bar{M}(k)}}\mathbf{x}(k)\|^2} \tag{6.74}$$

where

$$\mu(k) = \begin{cases} 1 - \frac{\bar{\gamma}}{|e(k)|} & \text{if } \mathbf{w}(k) \notin \mathcal{H}(k), \text{ i.e., if } |e(k)| > \bar{\gamma} \\ 0 & \text{otherwise} \end{cases} \tag{6.75}$$

In [17], a number of properties and an interesting geometrical interpretation of the SM-PU-NLMS algorithm update are provided, some of these results are discussed here. Fig. 6.9 depicts the situation where one coefficient updates out of three, i.e., $\bar{M} = 1$ and $N + 1 = 3$. As can be observed, the element $x_2(k)$ is the largest in magnitude among the elements of $\mathbf{x}(k)$, therefore a natural choice for $\mathbf{C}_{\mathcal{I}_{\bar{M}_2}(k)}$ is a diagonal matrix whose diagonal elements are $[0 \ 1 \ 0]$. The solution denoted by $\mathbf{w}_{\mathrm{SM-NLMS}}$ is obtained by an orthogonal projection starting from $\mathbf{w}(k)$ onto the closest boundary of the constraint set $\mathcal{H}(k)$. The angle denoted by θ shown in Fig. 6.9 is the angle between the direction of update $\mathbf{C}_{\mathcal{I}_{\bar{M}_2}(k)}\mathbf{x}(k) = [0 \ x_2(k) \ 0]^T$ and the input vector $\mathbf{x}(k)$. When $\bar{M}$ coefficients are updated, the general expression for the cosine of θ in $\mathbb{R}^{N+1}$ is given by the relation

$$\cos\theta = \frac{\|\mathbf{C}_{\mathcal{I}_{\bar{M}}(k)}\mathbf{x}(k)\|}{\|\mathbf{x}(k)\|} \tag{6.76}$$

whereas for the case in discussion, the particular expression for the cosine is

$$\cos\theta = \frac{|x_2(k)|}{\sqrt{|x_1(k)|^2 + |x_2(k)|^2 + |x_3(k)|^2}}$$

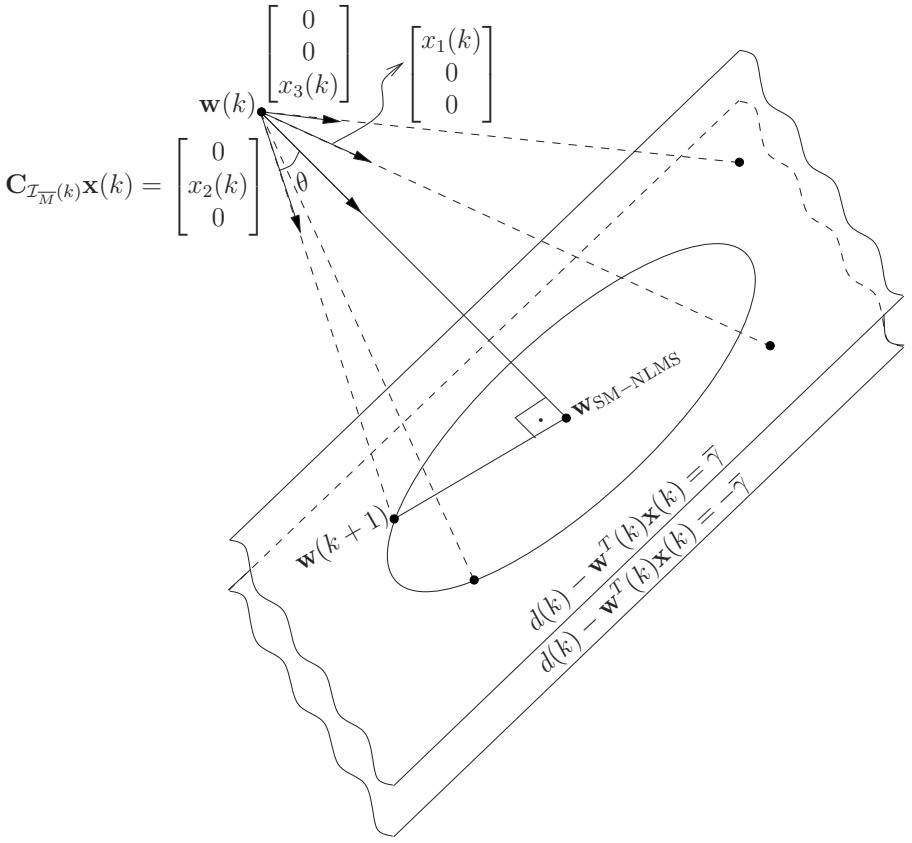


Figure 6.9 Partial coefficient update for $\mathbb{R}^3$ and $\bar{M} = 1$, where $|x_2(k)| > |x_1(k)| > |x_3(k)|$.

The SM-PUNLMS algorithm may face convergence problem whenever trying to find a solution in the constraint set. If the number of coefficients to be updated is small, $\cos\theta$ might become small according to equation (6.76), with θ becoming close to $\frac{\pi}{2}$, as can be observed in Fig. 6.10. As a result, the solution in the constraint set will depart from the SM-NLMS solution, and will give rise to stability problems.

A possible solution is to increase $\bar{M}$ up to the point where the solution provided by the SM-PUNLMS algorithm reaches a prescribed closer distance of SM-NLMS or NLMS solutions. Unfortunately this solution does not impose an upper bound on the value of $\bar{M}$, and it is highly probable that during initial iterations $\bar{M}$ would be close to the overall number of filter coefficients $N + 1$. On the other hand, it is desirable that $\bar{M} \ll N + 1$ in order to make the partial update effective in reducing the computational complexity.

Let's first define as $\bar{M}_{\max}$ the maximum number of coefficients that can be updated at any given iteration. It is now required to derive a strategy to control the number of coefficients to be updated while keeping a bound on the norm of the update. If $\|\mathbf{C}_{\mathcal{I}_{\bar{M}}(k)}\mathbf{x}(k)\|^2 = \|\mathbf{x}(k)\|^2$, it is straightforward

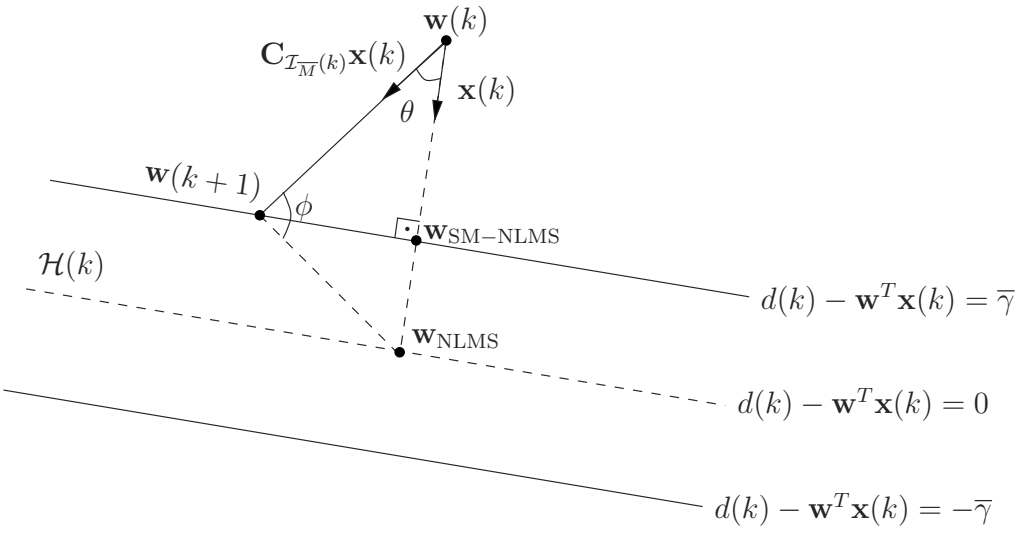


Figure 6.10 Projection in partial-update algorithms.

to verify that the angle ϕ is equal to $\frac{\pi}{2}$ and $\mathbf{w}_{\mathrm{NLMS}} - \mathbf{w}(k+1)$ represents the projection of $\mathbf{w}_{\mathrm{NLMS}} - \mathbf{w}(k)$ into $\mathbf{C}_{\mathcal{I}_{\bar{M}}(k)}\mathbf{x}(k)$. For angle $\phi < \frac{\pi}{2}$ the norm of the updating term might become large in order to meet the error modulus requirement, placing the partial solution far way from $\mathbf{w}_{\mathrm{NLMS}}$ and $\mathbf{w}_{\mathrm{SM-NLMS}}$. Indeed, whenever $\phi \geq \frac{\pi}{2}$ the norm of the updating term becomes smaller than the one required to turn the *a posteriori* error equal to zero (the one reaching $\mathbf{w}_{\mathrm{NLMS}}$). Then, an alternative solution is to increase the number of coefficients to update up to the condition that $\|\mathbf{C}_{\mathcal{I}_{\bar{M}}(k)}\mathbf{x}(k)\|^2 \geq \mu(k)\|\mathbf{x}(k)\|^2$, for $\mu(k) = 1 - \bar{\gamma}/|e(k)|$, or $\bar{M} = \bar{M}_{\max}$. This strategy will keep the angle ϕ lower bounded by $\frac{\pi}{2}$. If $\bar{M} = \bar{M}_{\max}$, increase the threshold $\bar{\gamma}$ temporarily at the kth iteration to

$$\bar{\gamma}(k) = \frac{(\|\mathbf{x}(k)\|^2 - \|\mathbf{C}_{\mathcal{I}_{\bar{M}(k)}}\mathbf{x}(k)\|^2)}{\|\mathbf{x}(k)\|^2}|e(k)| \tag{6.77}$$

Fig. 6.11 shows that this strategy temporarily expands the constraint set in order to allow a feasible solution in the case where the required number of coefficients to meet a solution in the constraint set exceeds $\bar{M}_{\max}$, at a given iteration.

Another possible strategy for the partial update is to choose the set of coefficients to be updated in a random manner [19] utilizing randomly partitions of the $N+1$ coefficients consisting of mutually exclusive subsets of $\bar{M}$ elements, each determined by the index set $\mathcal{I}_{\bar{M}}(k) = \{i_0(k) \ldots i_{\bar{M}-1}(k)\}$, as previously defined. This solution avoids the possible instability of the partial-update LMS algorithm originated by the choice of subsets in a deterministic manner, since in the latter case it is usually possible to generate input signals in which the algorithm fails to converge.

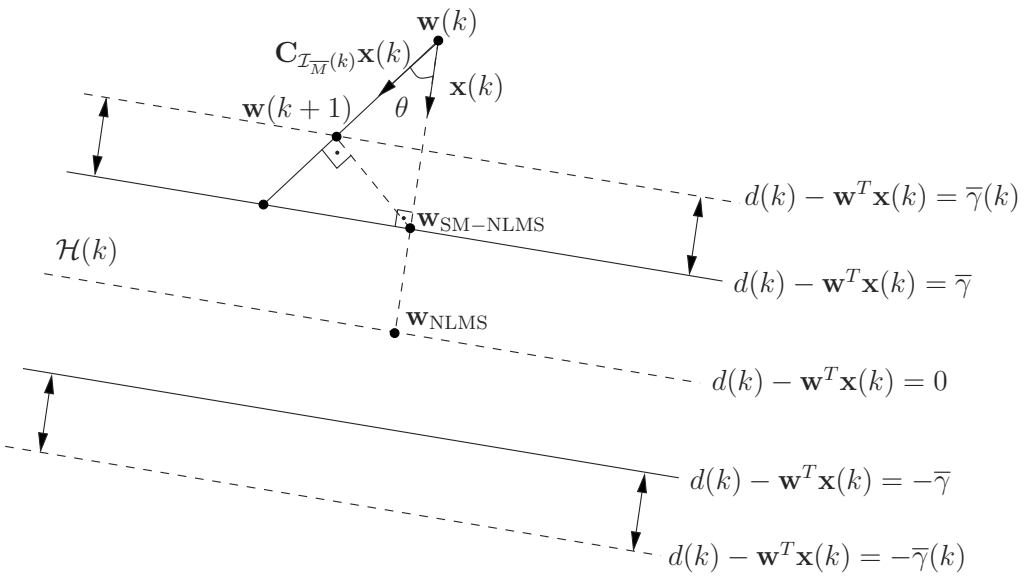


Figure 6.11 Variable constraint set $\mathcal{H}(k)$ with threshold $\bar{\gamma}(k)$.

6.9 SIMULATION EXAMPLES

In this section, some adaptive-filtering problems are described and solved by using some of the algorithms presented in this chapter.

Example 6.1: SM-AP Algorithms, system identification simulation

An adaptive-filtering algorithm is used to identify the system described in the example of subsection 3.6.2 using the following SM-AP algorithms:

- Set-membership affine projection using $L = 0$, $L = 1$ and $L = 4$.
- Set-membership partial-update affine projection with $\bar{M} = 5$, using $L = 0$, $L = 1$ and $L = 2$ and only for the eigenvalue spread of 20.

Do not consider the finite-precision case.

Solution:

All the results presented here for the affine projection and the SM-AP algorithms are obtained by averaging the results of 200 independent runs. We first run the affine projection algorithm with a value of $\mu = 0.18$, with $\gamma = 10^{-6}$. With this value of μ, the misadjustment of the affine projection

algorithm is about the same as that of the LMS algorithm with $\mu = 0.0128$ and eigenvalue spread of the input signal autocorrelation matrix of 20, see Table 3.1. Fig. 6.12 illustrates the learning curves for the eigenvalue spread 80 and distinct values of L. As expected the convergence speed and the misadjustment increase with the value of L.

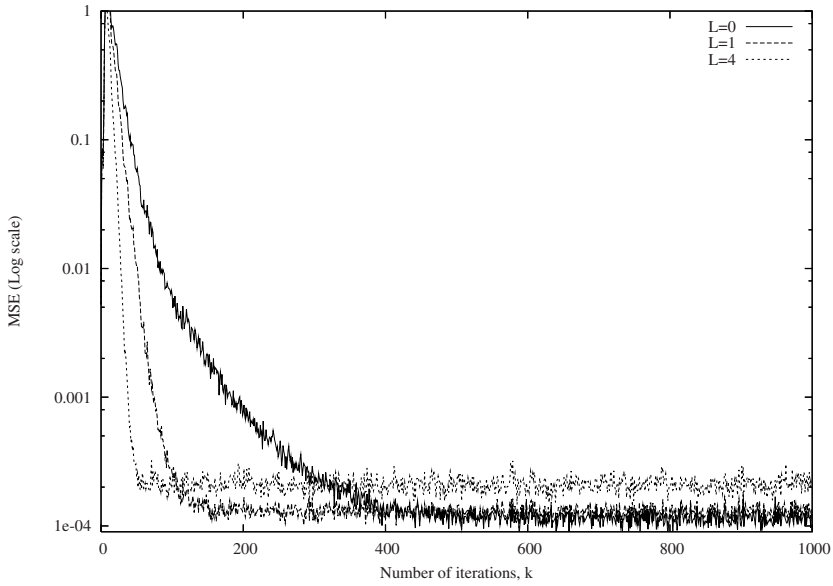


Figure 6.12 Learning curves for the affine projection algorithms for $L = 0$, $L = 1$, and $L = 4$, eigenvalue spread equal 80.

Table 6.2 lists the measured misadjustments along with their theoretical values obtained from equation (4.125) for distinct versions of the affine projection algorithms. As expected the misadjustment increases with the values of the reuse factor and with the ratio $\frac{\lambda_{\max}}{\lambda_{\min}}$.

Table 6.2 Evaluation of the Affine Projection Algorithm, $\mu = 0.18$

$\frac{\lambda_{\max}}{\lambda_{\min}}$	**Misadjustment,** $L = 0$		**Misadjustment,** $L = 1$		**Misadjustment,** $L = 4$	
	Experiment	**Theory**	**Experiment**	**Theory**	**Experiment**	**Theory**
1	0.1275	0.0989	0.2665	0.1978	0.9554	0.4945
20	0.1458	0.0989	0.2951	0.1978	1.0881	0.4945
80	0.1708	0.0989	0.3157	0.1978	1.2091	0.4945

Figs. 6.13, 6.14 and 6.15 depict the the learning curves for the simplified SM-AP algorithm for the eigenvalue spreads 1, 20 and 80, respectively. In each figure, distinct values of L are tested and the value of $\bar{\gamma}$ is $\sqrt{5}\sigma_n$. As can be observed, the convergence speed and the misadjustment increase with

the value of L. As will be discussed, a reduction in the misadjustment is achieved at the expense of mild increase in number of iterations for convergence.

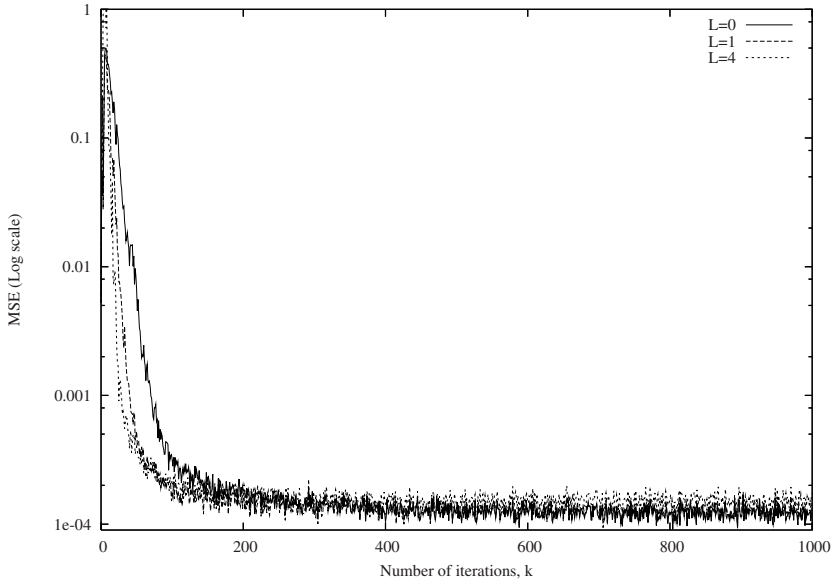


Figure 6.13 Learning curves for the SM-AP algorithms for $L = 0$, $L = 1$, and $L = 4$, eigenvalue spread equal 1.

Table 6.3 illustrates the convergence speeds of the affine projection algorithms and the SM-AP algorithms for distinct input signal eigenvalue spreads and distinct reuse factors. As can be observed, the SM-AP algorithms have convergence speeds comparable to the corresponding affine projection algorithms, being better for low values of L and worse for high values of L. The number of iterations for convergence is measured whenever the average square error reaches a value 5% above the noise floor.

Table 6.3 Convergence Speed of the Affine Projection and SM-AP Algorithms

	Convergence speed					
$\frac{\lambda_{\max}}{\lambda_{\min}}$	$L = 0$		$L = 1$		$L = 4$	
	AP	SM-AP	AP	SM-AP	AP	SM-AP
1	316	227	213	225	143	201
20	465	344	195	227	137	200
80	644	468	197	229	135	200

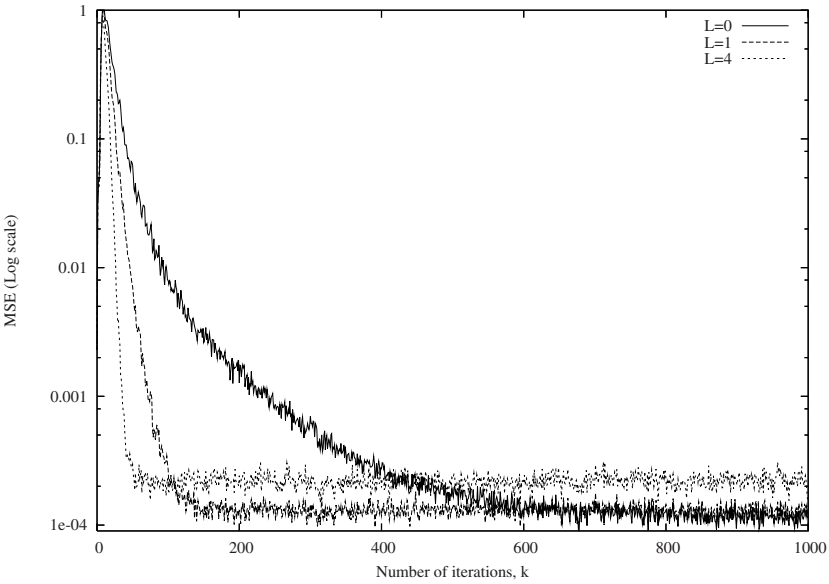


Figure 6.14 Learning curves for the SM-AP algorithms for $L = 0$, $L = 1$, and $L = 4$, eigenvalue spread equal 20.

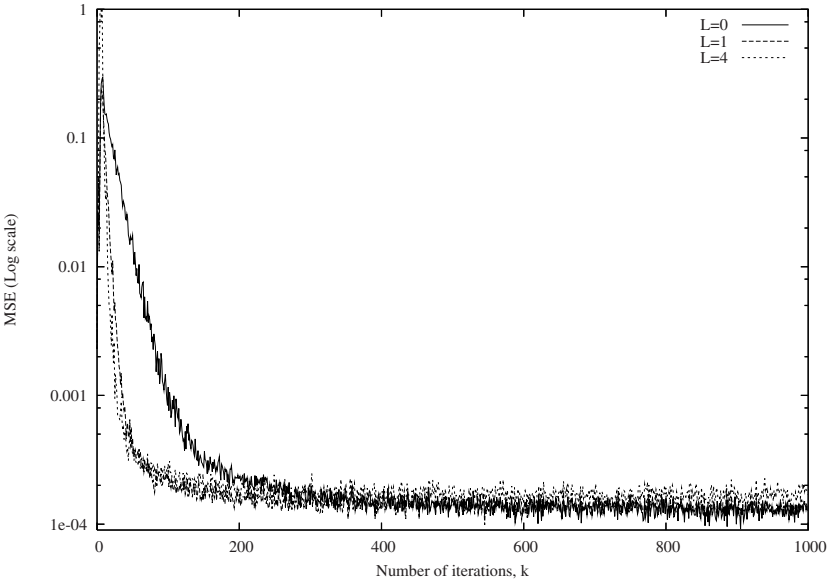


Figure 6.15 Learning curves for the SM-AP algorithms for $L = 0$, $L = 1$, and $L = 4$, eigenvalue spread equal 80.

Table 6.4 includes the measures misadjustments of the affine projection algorithms and the SM-AP algorithms considering the same input signal eigenvalue spreads and distinct reuse factors as before. As can be seen, the SM-AP algorithms have lower misadjustments than the corresponding affine projection algorithms for higher values of L.

Table 6.4 Misadjustment of the Affine Projection and SM-AP Algorithms

	Misadjustment					
$\frac{\lambda_{\max}}{\lambda_{\min}}$	$L=0$		$L=1$		$L=4$	
	AP	**SM-AP**	AP	**SM-AP**	AP	**SM-AP**
1	0.1275	0.1542	0.2665	0.1797	0.9554	0.3570
20	0.1458	0.2094	0.2951	0.2793	1.0881	0.5462
80	0.1708	0.2723	0.3157	0.3895	1.2091	0.6934

The SM-PUAP algorithm was set to update only 5 coefficients per iteration. For the SM-PUAP algorithm the learning curves are depicted in Fig. 6.16 for distinct values of L. The values of $\bar{\gamma}$ for $L = 0, 1$, and 2 are $\sqrt{5\sigma_n^2}$, $\sqrt{7\sigma_n^2}$, and $\sqrt{17\sigma_n^2}$, respectively. The corresponding measured misadjustments were 0.1979, 0.3137, and 0.8189. An efficient algorithm for the best selection of the updating coefficients in the partial-updating affine projection algorithm is an open problem, although some approximate solutions exist [21]. The choice of the coefficients to be updated relies on a low complexity procedure to sort out the $\bar{M}$ columns of $\mathbf{X}_{\text{ap}}^T(k)$ consisting of choosing the ones whose Euclidean norm have higher values.

□

6.9.1 Echo Cancellation Environment

The elimination of echo signals in communication networks and in hands-free communication environment are challenging problems in which adaptive filtering plays a major role [29]-[30].

The network echo, also known as line echo, is caused by the hybrid transformer whose main task is to convert the two-wire loop connection between the end user and the central office into a four-wire circuit. In the two-wire case, the signal in both directions traverses the two wires, whereas in the four-wires the signals in each direction are separated. Fig. 6.17 illustrates a very simplified long-distance telephone system where the location of the echo canceller is also included. The four-wire circuit exists only in long-distance connections and the delay included in Fig. 6.17 accounts for the traveling time of the signal from one hybrid to the other. Usually the far-end hybrid leaks back to the phone its own transmitted signal giving rise to the echo. If the echo delay is over 100 ms, its effect in the conversation is very disturbing. The early solution comprised of echo suppressor, whose aim was removing the echo from the talker by cutting off the outgoing hybrid port whenever an

incoming signal is detected. This approach works well for low round trip delays, but for large delays an adaptive echo canceller is more effective.

Echo cancellers are also used in acoustics echo cancellation problems where its task is to model the transfer function from the loudspeaker to the microphone in a given room. This application is more challenging than the network echo cancellation since the echo path impulse response is much longer, usually well above 500 taps, and changes quite rapidly. As depicted in Figs. 6.17 and 6.18, the echo cancellation problems in networks and in acoustics are closely related, with the latter requiring more sophisticated algorithms such as the subband adaptive filters of Chapter 12.

For both applications two measures of performance are the echo return loss (ERL) and the the echo return loss enhancement (ERLE). The ERL is ratio of the returned-echo power and the input-signal power, measuring the attenuation faced by the signal in the echo path. The ERL, measured in dB, is defined as

$$\text{ERL} = -10 \log \frac{\sigma_d^2}{\sigma_x^2} = -10 \log \frac{E[d^2(k)]}{E[x^2(k)]} \tag{6.78}$$

The ERLE measures the reduction in the echo obtained by utilizing the echo canceller, that is

$$\text{ERLE} = -10 \log \frac{\sigma_e^2}{\sigma_d^2} = -10 \log \frac{E[e^2(k)]}{E[d^2(k)]} \tag{6.79}$$

For simulation purposes we will utilize the models recommended by the International Telecommunication Union (ITU) in the ITU-T recommendation G.168 for digital network echo cancellers

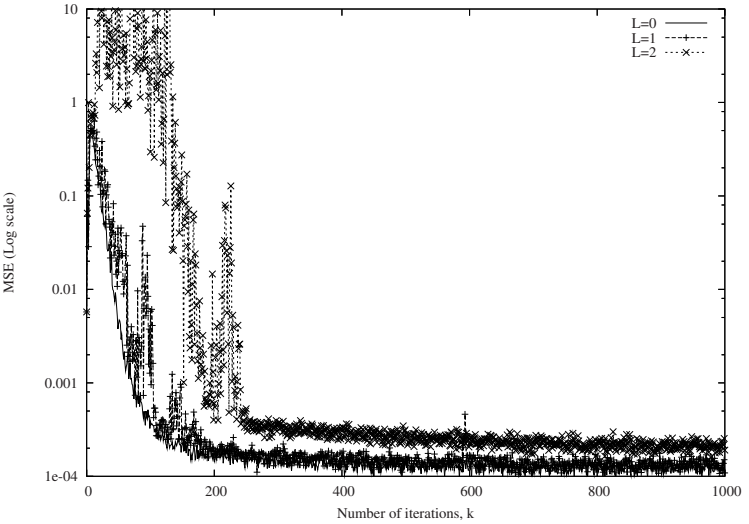


Figure 6.16 Learning curves for the SM-PUAP algorithms for $L = 0$, $L = 1$, and $L = 2$, eigenvalue spread equal 20.

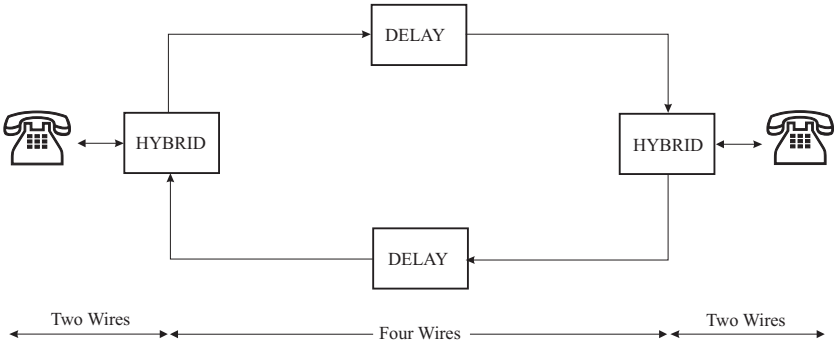


Figure 6.17 Two-wire to four-wire conversion in long-distance telephone calls.

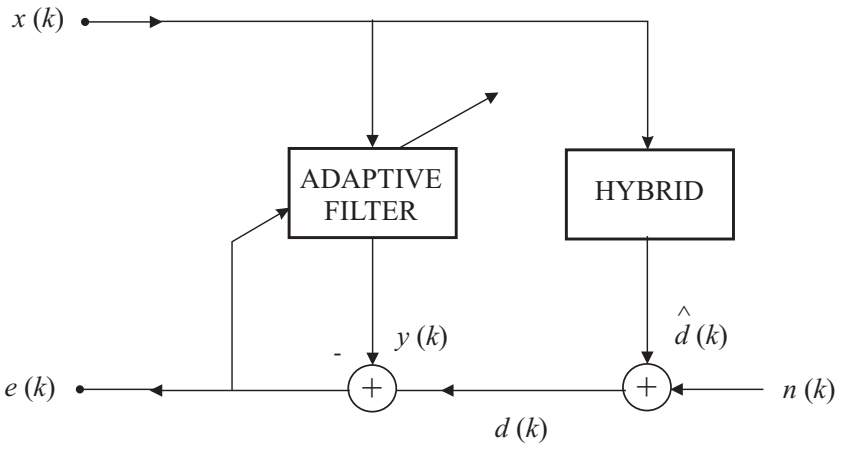
[31]. The main focus is to highlight the typical artificial input signals and echo path models utilized to test the performance of the echo canceller algorithms. The echo cancellers should be disabled during signaling transmission periods, however, no mention is given here to this and many other practical issues described in the recommendation related to the actual implementation of the echo canceller, see [31] for details.

The tests recommended by the standard ITU-T G.168 utilize particular signals such as noise, tones, facsimile signals and a set of composite source signals (CSS). In our simulations we apply the CSS input signal as input to the echo cancellers. The CSS simulates speech characteristics in single talk and double talk enabling a performance test for echo cancellers for speech signals. The CSS consists of speech signal, non speech signal and pauses. The speech signal activates the speech detectors and has approximately 50 msec of duration. The speech signal is followed by a pseudo-noise signal having constant magnitude Fourier transform whose phase changes with time. The pause is the third component of the CSS representing an amplitude modulation to the CSS and the usual pauses during a conversation. The pause duration ranges from 100 msec to 150 msec. Fig. 6.19 illustrates the CSS for single talk. The specific timings are:

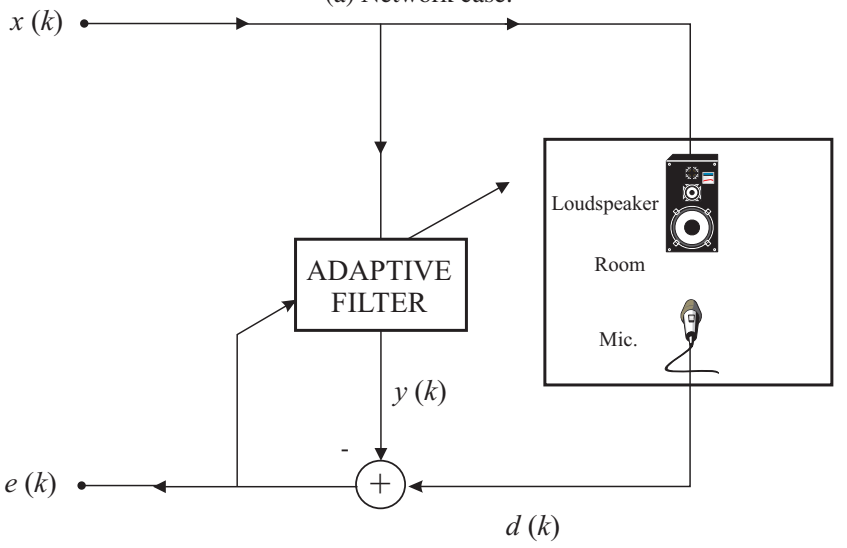
- Tvst (Speech signal): 48.62 msec.
- Tpn (Pseudo noise): 200.00 msec.
- Tpst (Pause): 101.38 msec.
- Tst1 (Half period): 350.00 msec.
- Tst (Full period): 700.00 msec.

The echo path model according to the recommendation ITU-T G.168, is a linear digital filter whose impulse response $h(k)$ is given by

$$h(k) = (K_i 10^{-\mathrm{ERL}/20}) m_i(k - \delta) \tag{6.80}$$



(a) Network case.



(b) Acoustics echo cancellation case.

Figure 6.18 Echo cancellation setups.

where ERL is the echo return loss defined in equation (6.78) and $h(k)$ consists of a delayed and attenuated version of any sequence sorted from $m_i(k), i = 1, 2, ..., 8$, for the channel models 1 to 8. These models represent channels whose origins range from hybrid simulation models to measured responses on telephone networks. The constants K_i are determined by the input signal used in the test [31] and are different for distinct echo path models.

Just for illustration Table 6.5 shows the sequence $m_1(k)$ composing the echo-path impulse response. In this case, for CSS type input signal, the scaling signal should be signal $K_1 = 1.39$ and the minimum value of the ERL to be used in the test is 6 dB. The resulting echo-path impulse response is depicted in Fig. 6.20. For the other cases, please refer to [31].

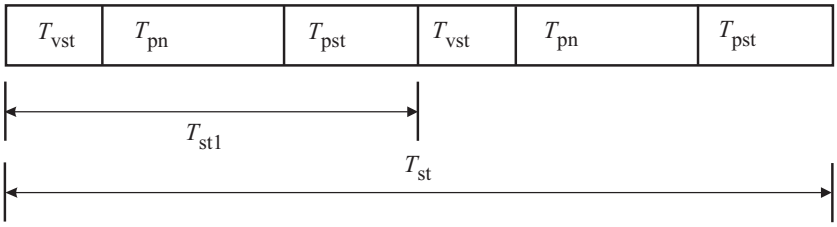


Figure 6.19 CSS single talk characteristics.

Table 6.5 Coefficients of $m_1(k)$, with k Ranging from 0 to 63, to be Read Column-wise

$m_1(k)$										
-0.00436	0.46150	0.00390	-0.03948	-0.01098	0.00745	0.01033	0.00899	0.00073	-0.00512	-0.00772
-0.00829	0.34480	-0.08191	-0.02557	-0.00618	0.00716	0.01091	0.00716	-0.00119	-0.00580	-0.00820
-0.02797	-10427	-0.01751	-0.03372	-0.00340	0.00946	0.01053	0.00390	-0.00109	-0.00704	-0.00839
-0.04208	0.09049	-0.06051	-0.01808	-0.00061	0.00880	0.01042	0.00313	-0.00176	-0.00618	-0.00724
-0.17968	-0.01309	-0.03796	-0.02259	0.00323	0.01014	0.00794	0.00304	-0.00359	-0.00685	
-0.11215	-0.06320	-0.04055	-0.01300	0.00419	0.00976	0.00831	0.00304	-0.00407	-0.00791	

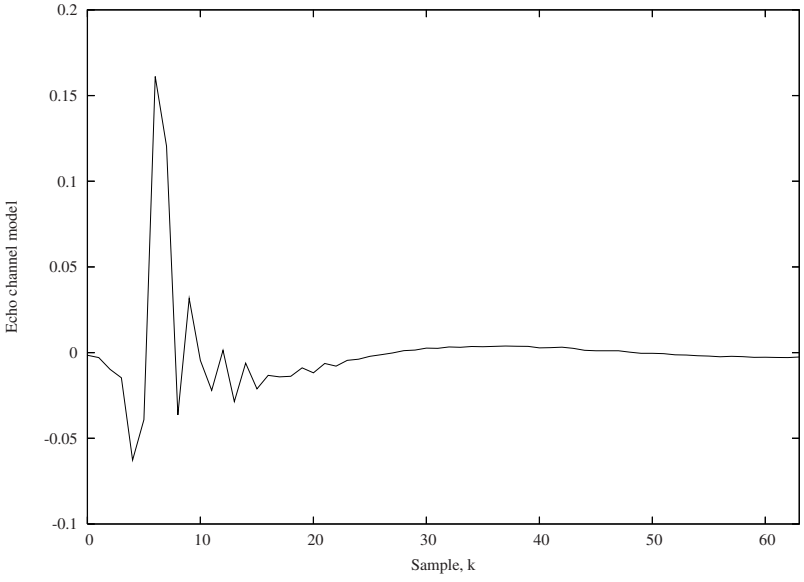


Figure 6.20 Echo-path impulse response.

Example 6.2: Echo cancellation simulations

For the algorithms pointed below, run simulations for an echo cancellation experiment consisting of 50 independent runs describing the average performance for single talk input signal for one of the eight channel models described in [31], specifically the one described in Table 6.5 with an $\mathrm{ERL} = 12$dBs. List the resulting ERLE in dB for each algorithm as well as their respective number of iterations to convergence, measured whenever the average of the last 100 error signals is 10% above the error in steady state. Utilize echo cancellers with sufficient order.

- Normalized LMS algorithm.
- RLS algorithm.
- SM-NLMS algorithm.
- The simplified SM-AP algorithm with $L = 0, 1, 4$.
- The SM-PUAP algorithm with $L = 0, 1, 4$, and $\bar{M} = \mathrm{floor}[\frac{2(N+1)}{3}]$ where $\mathrm{floor}[\cdot]$ indicates the largest integer smaller than $[\cdot]$.

For channel model 1, depict the learning curves for the simplified SM-AP and the SM-PUAP algorithms.

Solution:

The numbers and figures presented in this example are result of averaging fifty independent runs. The normalized LMS algorithm utilizes a value of $\mu = 0.5$, with the value of the regularization parameter of $\gamma = 10^{-6}$. The forgetting factor of the RLS algorithm is $\lambda = 0.99$. These values of μ and λ were chosen after some simulation trials indicating favorable performances of the corresponding algorithms. In the SM-AP algorithms distinct values of L are tested and the value of $\bar{\gamma}$ is 0.0002.

Fig. 6.21 depicts the CSS signal utilized in this example. Fig. 6.22 illustrates the error signal for the simplified SM-AP algorithm with $L = 0, 1, 4$, where it can be observed that the error reduces faster for the case with $L = 4$ since the algorithm is more sophisticate, even though the convergence speeds for $L = 1$ and $L = 4$ are quite similar. Fig. 6.23 shows that with the SM-PUAP algorithm the convergence speed is not substantially reduced, showing that the partial updating strategy is very useful. A low complexity way to choose the elements to be updated was to sort out the $\bar{M}$ columns of $\mathbf{X}_{\mathrm{ap}}^T(k)$ whose Euclidean norm have higher values. The SM-PUAP algorithm was set to update only $\frac{2}{3}$ of the coefficients.

Table 6.6 lists the relevant parameters in the echo cancellation environment, namely the ERLE in dB for each algorithm as well as their respective convergence speed. As can be seen in Table 6.6, the algorithms SM-NLMS ($L = 0$), SM-AP, and SM-PUAP require less updates than the remaining algorithms compared. The fastest converging algorithm is the SM-AP ($L = 4$) but it requires the highest computational complexity among the set-membership algorithms. The algorithms SM-AP and SM-PUAP, with $L = 4$, are faster converging than the RLS while requiring much less updates and

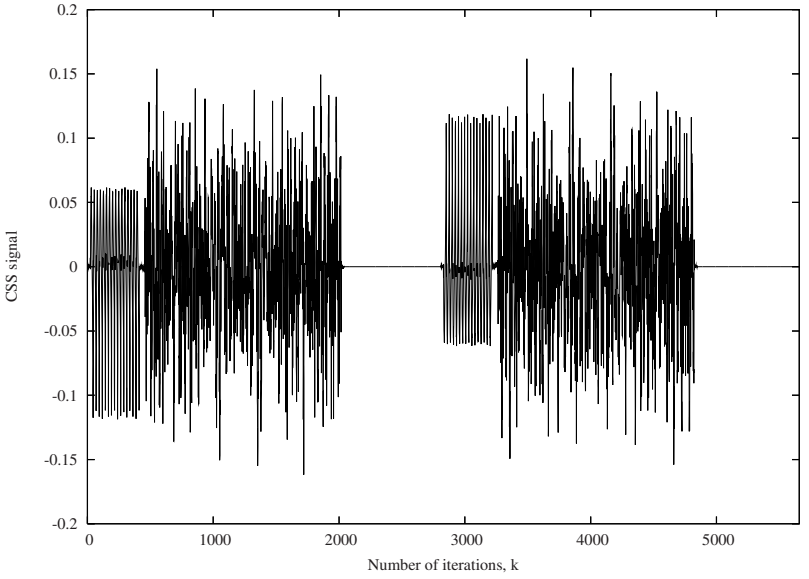


Figure 6.21 CSS signal.

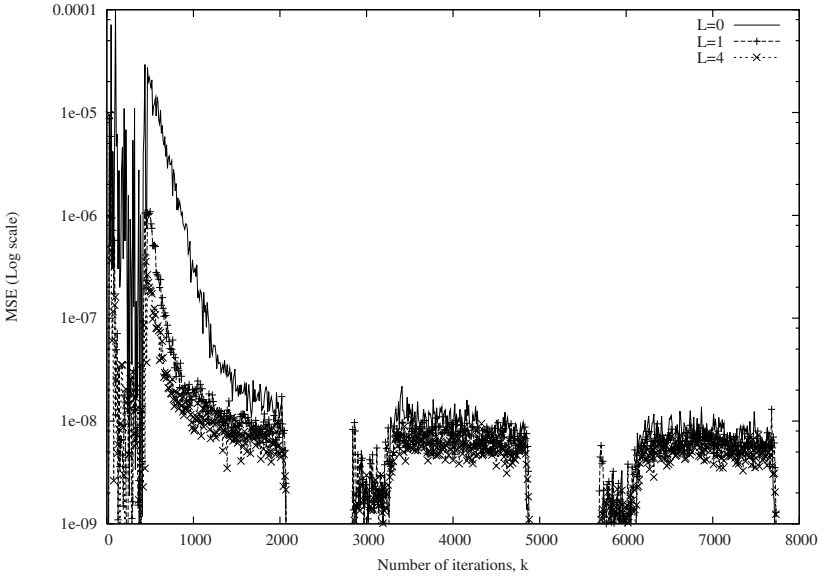


Figure 6.22 Learning curves for the simplified SM-AP algorithm with $L = 0, 1, 4$.

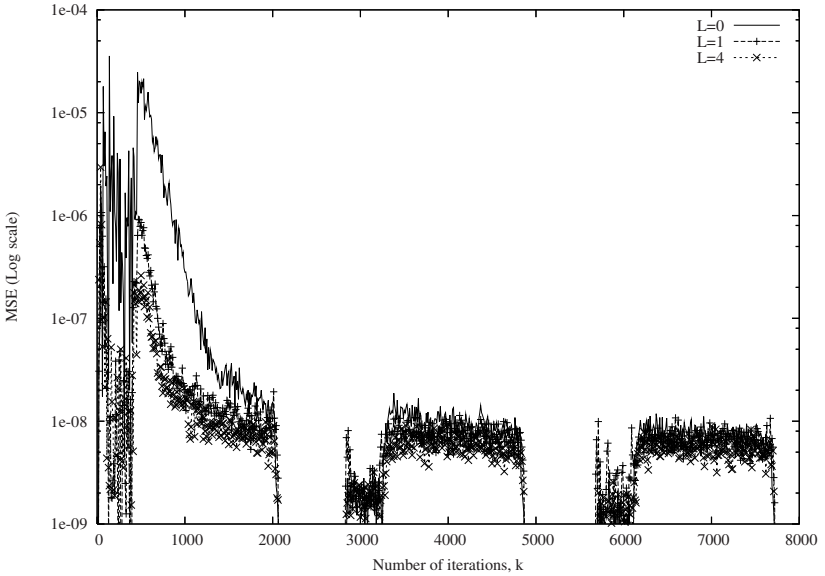


Figure 6.23 Learning curves for the SM-PUAP algorithm $L = 0, 1, 4$.

computations. On the other hand the RLS algorithm lead to much higher ERLE than the remaining algorithms followed by the NLMS. The SM-NLMS and NLMS algorithms have less computations but are slow converging as compared to the remaining SM-AP algorithms of this example.

□

6.9.2 Wireless Channel Environment

A typical continuous-time model for mobile communication channels is described by [33]

$$\tilde{h}(t) = \sum_{i=0}^{I} \sqrt{p_i} a_i(t) b_i(t - \tau_i) \tag{6.81}$$

where t is the time variable, p_i represents the power of the ith tap weight of the FIR model, $a_i(t)$ is complex Gaussian process that multiplies the corresponding transmitted symbol denoted by $b_i(t-\tau_i)$, and τ_i accounts for the relative delay that the receiver detects the ith replica of the transmitted symbols.

The power spectrum of $a_i(t)$ is responsible for the rate of fading of the ith replica (or reflection) of the transmitted symbols. This spectrum is also known as Doppler spectrum. The complete model requires the specification of the Doppler spectrum of the tap weights, denoted by $R_a(f)$ with f being the analog frequency, the delays τ_i, as well as the powers p_i, for $i = 0, \ldots, I$.

Table 6.6 Simulation Results: **Channel Model - ITU-T G.168, No. 1**

	Reuse Factor L		
	0	1	4
	Updates		
SM-AP	1320	497	290
SM-PUAP	1335	584	364
	Convergence		
NLMS	8423	-	-
RLS	6598	-	-
SM-AP	2716	2289	1832
SM-PUAP	2725	2303	1832
	ERLE		
NLMS	80.30	-	-
RLS	307.83	-	-
SM-AP	42.96	43.00	43.62
SM-PUAP	43.87	42.72	43.42

The process $a_i(t)$ is the result of a cluster of multipath components that cannot be resolved by the receiver, arriving within a range of delays[3]. Usually for outdoor environments of mobile communication systems, the model for the Doppler power spectrum is represented by the Jakes model [32] given by

$$R_a(f) = \begin{cases} \frac{1}{\pi f_{\mathrm{D}}} \frac{1}{\sqrt{1-(\frac{f}{f_{\mathrm{D}}})^2}} & \text{for} \quad |f| \leq f_{\mathrm{D}} \\ 0 & \text{for} \quad |f| > f_{\mathrm{D}} \end{cases}$$

where $f_{\mathrm{D}} = \frac{v}{\lambda_s} = \frac{v f_o}{c}$ is the maximum Doppler frequency shift, λ_s is the carrier wavelength, v is the mobile velocity in m/s, c is the speed of light (3.00×10^8 m/s), and f_o is the carrier central frequency.

If we assume that the input signal is complex and bandlimited to a bandwidth around BW, the received signal can be generated by filtering the input signal through a tapped delay line whose tap coefficients are given by $\sqrt{p_i}a_i(t)$ and the delay elements correspond to $T = \frac{1}{\mathrm{BW}}$ [33].

As an illustration, Table 6.7 lists the parameters of test channel models for an outdoor to indoor environment to be utilized in simulations. These models originate from a standard described in [34] for the Universal Mobile Telecommunications System (UMTS). In Table 6.7, the delays are relative

[3] $\tau_i - \frac{1}{2\mathrm{BW}} < \tau < \tau_i + \frac{1}{2\mathrm{BW}}$ with BW denoting the bandwidth of the transmitted signal.

to the first tap whereas the power is relative to the strongest tap. The Doppler power spectrum applies to each tap.

Let's consider for illustration a typical case of UMTS where the chip duration is 260.04 nano seconds (ns) for a transmission rate of 3.84 Mc/s (Mega chips per second). In the case the time difference between two multipath components is at least $260.04 = \frac{1}{3.84}$ ns, it is possible for the receiver to separate them. For example according to Table 6.7, in a digital simulation environment where the input signal is sampled at chip rate for channel B, it will be possible to verify the presence of the multipath signals of the taps at approximately

- 1 chip from the reference for tap 2.
- 3 chips from the reference for tap 3.
- 5 chips from the reference for tap 4.
- 9 chips from the reference for tap 5.
- 14 chips from the reference for tap 6.

where it was taken into consideration that the relative delays in the table represent the time where the energy of the continuous-time reflection reaches its maximum.

Table 6.7 Channel Model Parameters: Outdoor to Indoor Test Environment with Jakes Doppler Spectrum

Tap	**Channel A**		**Channel B**	
	Relative Delay	**Average Power**	**Relative Delay**	**Average Power**
	ns	**dB**	**ns**	**dB**
1	0	0	0	0
2	110	-9.7	200	-0.9
3	190	-19.2	800	-4.9
4	410	-22.8	1200	-8.0
5	–	–	2300	-7.8
6	–	–	3700	-23.9

The coefficients of a time-varying channel including the Doppler effects can be generated as depicted in Fig. 6.24, where $\bar{n}(t)$ is a Gaussian noise source and the Doppler filter is an approximation of $H_{\mathrm{D}}(f) = \sqrt{R_a(f)}$. Fig. 6.25 shows an efficient way to generate the coefficients of the channel model [36]-[37], where from two real-valued Gaussian sources with $\bar{N}+1$ points we calculate their symmetrical FFT spectrum [38]. Then we multiply the FFT outputs by $H_{\mathrm{D}}(f_{\bar{m}})$ where $f_{\bar{m}} = \bar{m}\frac{2f_{\mathrm{D}}}{\bar{N}+1}$ for $\bar{m} = 0, 1, \ldots, \bar{N}$, and the resulting vector is applied as input to an $\bar{N}+1$ length IFFT. The quadrature and in-phase results are squared at each point in time, added, with the result square rooted. Finally, an $\bar{N}+1$-length time series is generated. In an actual simulation environment the

Gaussian noise is generated with around α_1 samples per period of the maximum Doppler frequency, that is $\frac{1}{\alpha_1 f_{\mathrm{D}}}$, therefore the sampling rate of the channel coefficients is around $\alpha_1 f_{\mathrm{D}}$ with α_1 being an integer usually chosen in the range 5 to 12. As can be noticed the coefficients of the channel model are generated from the Jakes model of the Doppler effect. However, the system simulation takes place at much higher frequency rate denoted as f_{sim}. As a result, an interpolation operation given by $L_{\mathrm{sim}} = \mathrm{floor}[\frac{f_{\mathrm{sim}}}{\alpha_1 f_{\mathrm{D}}}]$ should be applied to the coefficients of the channel model.

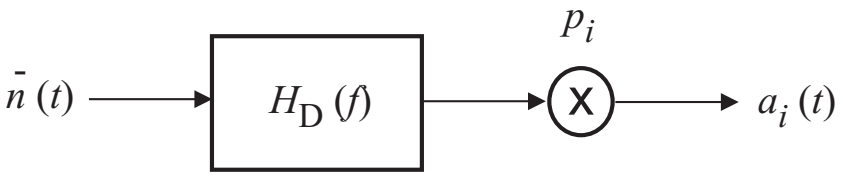


Figure 6.24 Generation of multipath coefficient including Doppler effect.

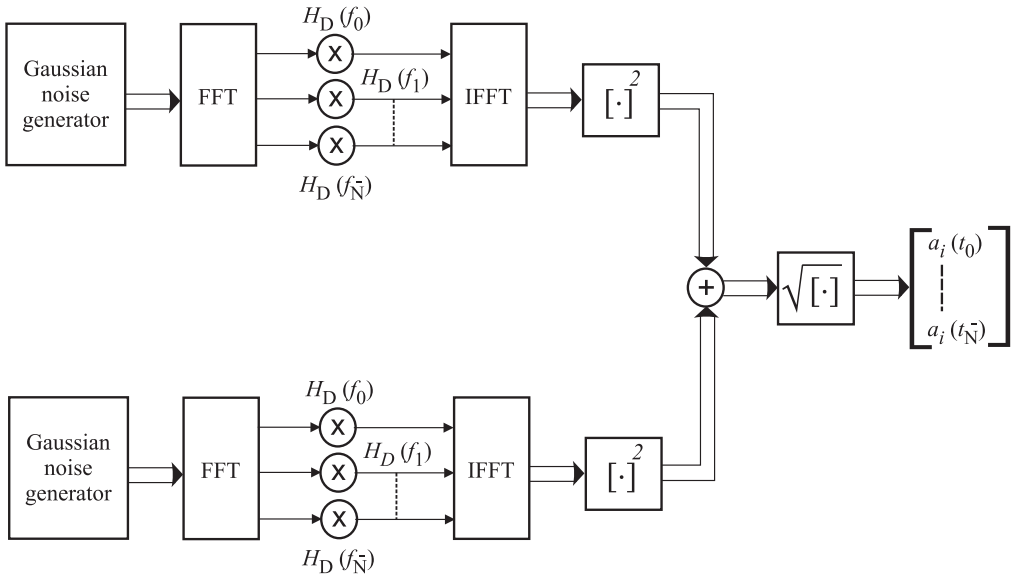


Figure 6.25 Simulation setup for Jakes model.

Example 6.3: CDMA receiver simulations

Consider a downlink connection of a synchronous direct-sequence code-division multiple access (DS-CDMA) system with J users, $G+1$ chips per symbol and a channel with $I+1$ paths. Assume the user receiver is moving at $v = 30.00$m/s and the carrier frequency is at $f_o = 1.0$GHz. We consider a simple model for the channel inspired by the UMTS test model above described. The channel model should be generated at a simulation sampling rate of at least $f_{\mathrm{sim}} = \alpha_2 \times \frac{1}{T} = \alpha_2 \mathrm{BW}$ samples per second, with α_2 being normally an integer ranging from 5 to 12. It is worth emphasizing again that the channel coefficients will be generated at a much lower rate than the simulation sampling

rate. As a result, some standard interpolation technique [38] should be used in order to match the channel model generation rate with simulation sampling rate.

Consider that the chip rate of the CDMA system is 0.5 Mc/s (Mega chips per second) and that we utilize $\alpha_2 = 10$ samples per chip to simulate the system. As such, the CDMA system simulation has sampling rate equal to 5 Msamples/s. In this case the interpolation factor applied to the chip level signal should be $L_{\text{interp}} = \text{floor}[\frac{5\text{Mc/s}}{\frac{1}{T}}] = \text{floor}[\frac{5\text{Mc/s}}{\text{BW}}]$, where $\text{floor}[\cdot]$ indicates the largest integer smaller than $[\cdot]$. The sampling frequency that the channel model should be generated is then given by

$$f_{\text{sim}} \approx \text{BW} L_{\text{interp}} \approx \alpha_1 f_{\text{D}} L_{\text{sim}}$$

in Msamples/s. In this particular discussion, assuming the input signal sampling rate equal to the chip rate the interpolation factor L_{interp} is equal to 10. Note that in the above discussion we satisfy the sampling theorem by utilizing the complex envelope, that is the complex lowpass equivalent, of the transmitted signal. This means the sampling rate is equal to the channel bandwidth.

Assuming the channel model as described is constant during each symbol interval, and that the input signal vector $\mathbf{x}(k)$ is given by[4]

$$\begin{aligned}\mathbf{x}(k) &= \sum_{j=1}^{J} A_j b_j(k) \mathbf{C}_j \mathbf{h}(k) + \mathbf{n}(k) \\ &= \sum_{j=1}^{J} A_j b_j(k) \mathbf{C}_j \mathbf{h}(k) + \mathbf{n}_{\text{n}}(k) + \mathbf{n}_{\text{ISI}}(k)\end{aligned} \tag{6.82}$$

where $\mathbf{x}(k)$ is an $(N+1 = G+I+1) \times 1$ vector and $\mathbf{n}_{\text{n}}(k)$ is defined in equation (6.58). We consider that $\mathbf{n}_{\text{n}}(k)$ is a complex Gaussian noise vector with $E[\mathbf{n}_{\text{n}}(k)\mathbf{n}_{\text{n}}^H(k)] = \sigma_n^2 \mathbf{I}$. The symbols $b_j(k)$ are four QAM given by $\frac{\sqrt{2}}{2}\{\pm 1 \pm j1\}$, where the amplitude of user j is A_j. The channel vector is $\mathbf{h}(k) = [h_0(k) \ \ldots \ h_I(k)]^T$ and the $(N+1) \times (I+1)$ convolution matrix $\mathbf{C}_k$ contains one-chip shifted versions of the signature sequence for user j given by $\mathbf{s}_j = [s_{j,0} \ \ s_{j,1} \ldots s_{j,G}]^T$. Matrix $\mathbf{C}_k$ has the following format

$$\mathbf{C}_j = \begin{bmatrix} s_{j,0} & 0 & 0 & \cdots & 0 \\ s_{j,1} & s_{j,0} & 0 & \cdots & 0 \\ s_{j,2} & s_{j,1} & s_{j,0} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & s_{j,G} & s_{j,G-1} & \cdots & s_{j,G-I} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & s_{j,G} \end{bmatrix} \tag{6.83}$$

This example aims to access the bit error rate (BER) performance of some adaptive-filtering algorithms such as:

[4] In an actual implementation $\mathbf{x}(k)$ originates from the received signal after filtering it through a chip-pulse matched filter and then sampled at chip rate.

- Normalized LMS algorithm.
- RLS algorithm.
- SM-NLMS algorithm.
- The simplified SM-AP algorithm with $L = 4$ and time-varying $\bar{\gamma}(k)$.
- The SM-PUAP algorithm with $L = 1$, and $\bar{M} = \text{floor}[\frac{(N+1)}{1.8}]$ where $\text{floor}[\cdot]$ indicates the largest integer smaller than $[\cdot]$.

The receiver of the DS-CDMA system simulation setup is depicted in Fig. 6.26 where we utilize as spreading sequences the Gold sequences of length $G + 1 = 7$ listed in Table 6.26 [35]. The Gold sequences are not orthogonal to each other leaving some multi-access interference from the other users in the CDMA system on the information of the user of interest, even in synchronous transmission case.

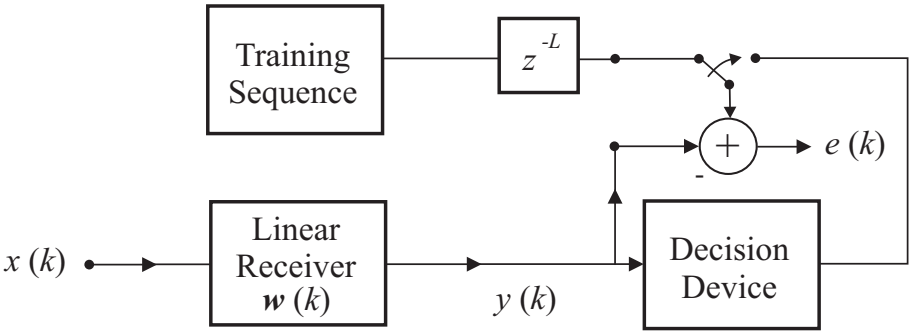


Figure 6.26 Simulations setup for DS-CDMA example.

Table 6.8 Length 7 Gold Sequences

Sequences	Gold Sequences						
s_1	1	0	0	1	0	1	1
s_2	1	1	1	0	1	0	0
s_3	0	1	1	1	1	1	1
s_4	1	1	1	0	0	0	1
s_5	1	0	1	0	1	1	0
s_6	0	0	0	0	1	0	1
s_7	1	1	0	1	1	0	0
s_8	0	0	1	1	0	0	0
s_9	0	1	0	0	0	1	0

All users are synchronized such that all their information face the same channel with three paths with relative powers given by 0, -0.9 and -4.9 dB, respectively. The relative delays between the paths are determined by a uniformly distributed random variable whose outcome is mapped to integers in the range 1 to 4, where these integers represent the number of chips.

The system starts with 5 users where all the 4 interferers have transmission powers 3 dB below the desired user power level. The corresponding signal to noise ratio, defined as the ratio between the desired user symbol energy per bit and the environment noise, is given by $E_b/N_0 = 20$ dB. The quantity $N_0/2$ corresponds to power spectral density of the noise for positive and negative frequencies, that is N_0 is average noise power per bandwidth where the noise is measured at the receiver input. At 2000 symbols, an interferer with the same power as the desired user power enters the system, whereas 2 interferers with the same level of power disconnect. This dynamic behavior aims at addressing, for this particular example, if some noticeable disturbance to the receiver performance originates from user access and disconnection from the system.

Plot the evolution of the estimated of the noise and ISI powers as compared with the actual interference power.

Solution:

For this example we measure the results by averaging the outcomes from 50 independent runs. In the case of the normalized LMS algorithm the value of μ is 0.3, whereas the regularization parameter value is $\gamma = 10^{-6}$. The RLS algorithm is implemented with $\lambda = 0.97$. Again these values of μ and λ were chosen after some simulation trials. The SM-AP algorithm uses $L = 4$ and variable $\bar{\gamma}$, whereas the SM-PUAP algorithm uses $L = 1$.

For a better view of the results the channel was allowed to change completely in an interval of 50 symbols, equivalent to 450 chips. Fig. 6.27 depicts the first 450 samples of the learning curves for the algorithms compared in this example, whereas Fig. 6.28 shows the behavior of these algorithms in the long run. In both figures the channel changes are noticeable every 350 chips, where the first change occurs at around 370 chips due to the channel plus spreading delays. As can be observed, the NLMS, RLS, SM-NLMS and the SM-AP algorithms were able to track the changes in the channel to some extent. However, as shown in Fig. 6.29, the simplified SM-PUAP algorithm with $L = 1$ using variable $\bar{\gamma}$ has very competitive performance since it is among the fastest converging in such nonstationary environment. Very similar result is obtained with the simplified SM-AP algorithm which has higher computational cost. All the algorithms did not show any noticeable distinct behavior after the entrance and exit of users in the system, mainly due to the fact that the channel model changes at every 50 symbols was the main source of changes.
Fig. 6.30 plots the evolution of the estimated noise and ISI powers as compared with the actual interference power. The estimated curve was obtained using equation (6.63) with $\alpha = 0.96$. As can be observed, the estimated average power of the interferences follows closely its actual value for this particular example, except at iteration 2000 when the interference estimate takes a few iterations to track the true interference power. The accurate estimate of the disturbances turns the SM-AP algorithms very attractive since virtually no environment dependent parameter is required to achieve good overall performance.

□

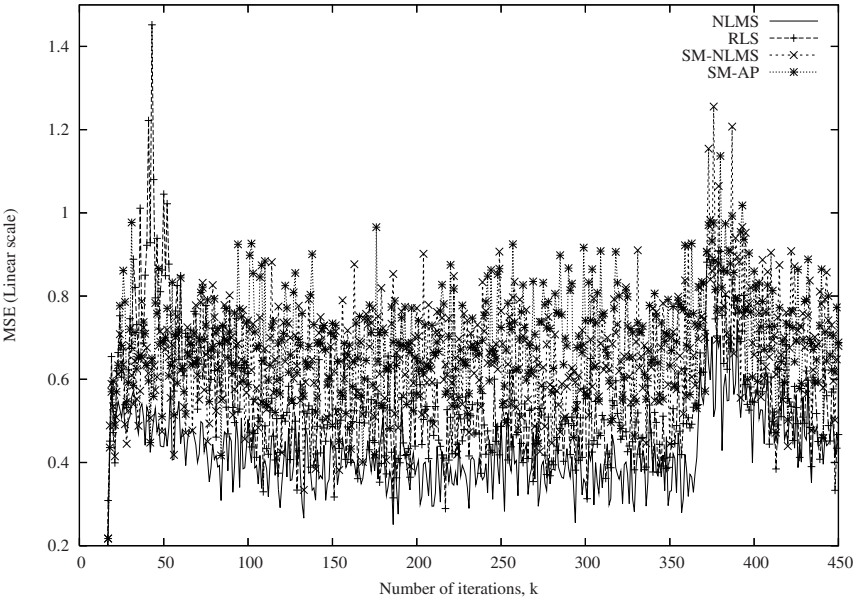


Figure 6.27 Learning curves for the NLMS, RLS, SM-NLMS, and SM-AP algorithms; 250 iterations.

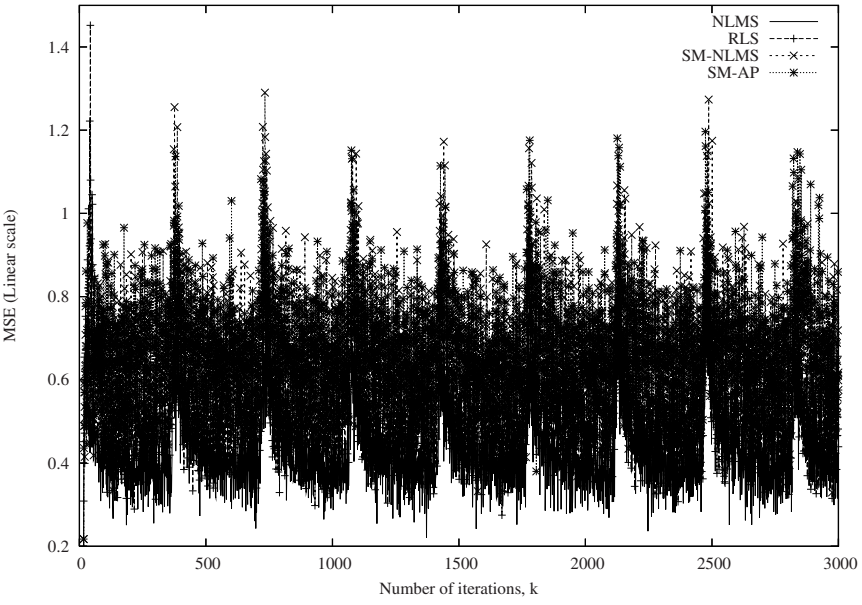


Figure 6.28 Learning curves for the NLMS, RLS, SM-NLMS, and SM-AP algorithms.

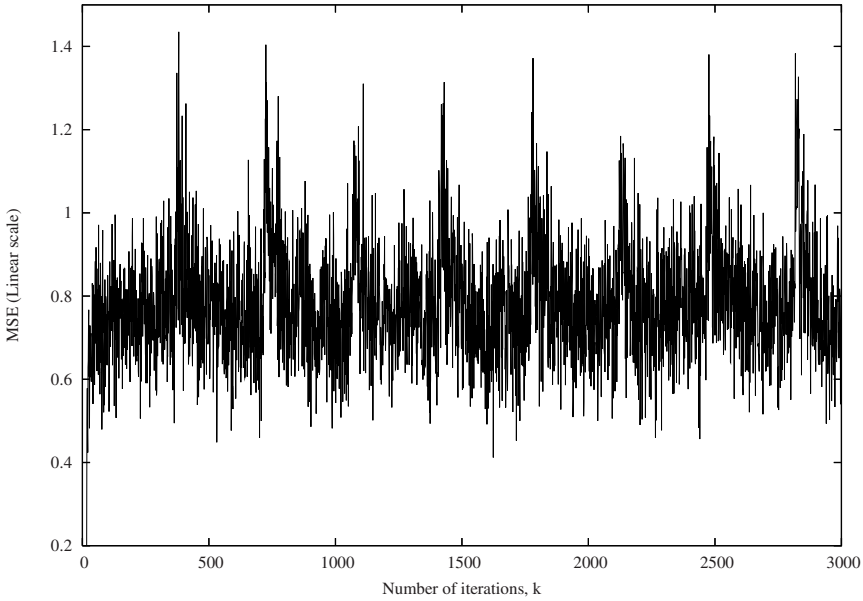


Figure 6.29 Learning curve SM-PUAP algorithms, $L = 1$.

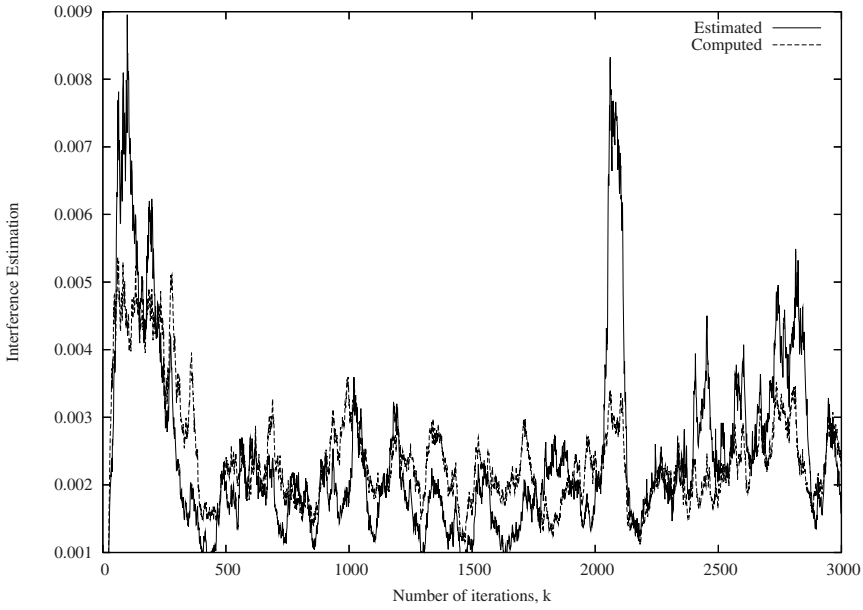


Figure 6.30 Interference estimation.

6.10 CONCLUDING REMARKS

In this chapter, a number of adaptive affine projection algorithms were derived utilizing the set-membership concept. Although the algorithms in the affine projection family might have high misadjustment, their combination with deterministic objective functions leading to data selective updating results in computationally efficient algorithms with low misadjustment and high convergence speed. The set-membership family of algorithms can be very attractive for mobile terminals, sensor arrays and embedded systems where by avoiding unnecessary computation the battery life is increased. In stationary environments, the set-membership algorithms require more frequent updating during the early iterations, as a consequence, if the computational complexity is of major concern some strategy to reduce even further the computation count is required. The proposed solution is to introduce the concept of partial update, in which only a subset of the adaptive filter are updated in each iteration. It is mentioned that some caution should be exercised in choosing the selection of the coefficients in order to avoid stability problems. The resulting set-membership affine projection algorithms with partial update are powerful solutions to exploit the tradeoff between speed of convergence and misadjustment with computational burden.

It should be mention that there are set-membership algorithms with automatic data-reusing factor according with the progress of the algorithm [20], [23]. Simulation results show that in most of iterations the SM-AP algorithm requires a small number of reuses, that is in the limit, it becomes the SM-NLMS or the SM-BNLMS algorithms. The set-membership technique can also be applied to generate constrained affine projection algorithms with low computational complexity as proposed in [39].

6.11 REFERENCES

1. F. C. Schweppe, "Recursive state estimate: Unknown but bounded errors and system inputs," *IEEE Trans. on Automatic Control*, vol. 13, pp. 22-28, Feb. 1968.

2. E. Fogel and Y.-F. Huang, "On the value of information in system identification - bounded noise case," *Automatica*, vol. 18, pp. 229-238, March 1982.

3. J. R. Deller, "Set-membership identification in digital signal processing," *IEEE Acoust., Speech, Signal Processing Magazine*, vol. 6, pp. 4-20, Oct. 1989.

4. J. R. Deller, M. Nayeri, and S. F. Odeh, "Least-squares identification with error bounds for real-time signal processing and control," *Proceedings of IEEE*, vol. 81, pp. 815-849, June 1993.

5. S. Dasgupta and Y.-F. Huang, "Asymptotically convergent modified recursive least-squares with data dependent updating and forgetting factor for systems with bounded noise," *IEEE Trans. on Information Theory*, vol. 33, pp. 383-392, May 1987.

6. A. K. Rao and Y.-F. Huang, "Tracking characteristics of an OBE parameter-estimation algorithm," *IEEE Trans. on Signal Processing*, vol. 41, pp. 1140-1148, March 1993.

7. S. Gollamudi, S. Nagaraj, S. Kapoor, and Y.-F. Huang, "Set-membership filtering and a set-membership normalized LMS algorithm with an adaptive step size," *IEEE Signal Processing Letters*, vol. 5, pp. 111-114, May 1998.

8. S. Gollamudi, S. Nagaraj, S. Kapoor, and Y.-F. Huang, "Set-membership adaptive equalization and updator-shared implementation for multiple channel communications systems," *IEEE Trans. on Signal Processing*, vol. 46, pp. 2372-2384, Sept. 1998.

9. S. Nagaraj, S. Gollamudi, S. Kapoor, and Y.-F. Huang, "BEACON: An adaptive set-membership filtering technique with sparse updates," *IEEE Trans. on Signal Processing*, vol. 47, pp. 2928-2941, Nov. 1999.

10. P. S. R. Diniz and S. Werner, "Set-membership binormalized data reusing LMS algorithms," *IEEE Trans. on Signal Processing*, vol. 51, pp. 124-134, Jan. 2003.

11. S. Werner and P. S. R. Diniz, "Set-membership affine projection algorithm," *IEEE Signal Processing Letters*, vol. 8, pp. 231-235, Aug. 2001.

12. A. Antoniou and W.-S. Lu, *Practical Optimization: Algorithms and Engineering Applications*, Springer, New York, NY, 2007.

13. J. A. Apolinário, M. L. R. de Campos, and P. S. R. Diniz, "The binormalized data-reusing LMS algorithm," *IEEE Trans. on Signal Processing*, vol. 48, pp. 3235-3242, Nov. 2000.

14. S. C. Douglas, "Adaptive filters employing partial updating," *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 44, pp. 209-216, March 1997.

15. T. Aboulnasr and K. Mayyas, "Complexity reduction of the NLMS algorithm via selective coefficient updating," *IEEE Trans. on Signal Processing*, vol. 47, pp. 1421-1427, May 1999.

16. S. C. Douğançay and O. Tanrikulu, "Adaptive Filtering algorithms with selective partial updating," *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, pp. 762-769, Aug. 2001.

17. S. Werner, M. L. R. de Campos, and P. S. R. Diniz, "Partial-update NLMS algorithm with data-selective updating," *IEEE Trans. on Signal Processing*, vol. 52, pp. 938-949, April 2004.

18. S. Werner, M. L. R. de Campos, and P. S. R. Diniz, "Mean-squared analysis of the partial-update NLMS algorithm," *Brazilian Telecommunications Journal - SBrT*, vol. 18, pp. 77-85, June 2003.

19. M. Godavarti and A. O. Hero, III, "Partial update LMS algorithms," *IEEE Trans. on Signal Processing*, vol. 53, pp. 2384-2399, July 2005.

20. S. Werner, J. A. Apolinário Jr., and P. S. R. Diniz, "Set-membership proportionate affine projection algorithms," *EURASIP- Journal on Audio, Speech, and Music Processing*, vol. 2007, pp. 1-10, Article ID 34242, 2007.

21. S. C. Douğançay and O. Tanrikulu, "Selective-partial-update NLMS and affine projection algorithms for acoustic echo cancellation," *Proc. IEEE Intern. Conf. on Acoust. Speech and Signal Processing*, Istanbul, Turkey, pp. 448-451, June 2000.

22. P. S. R. Diniz, R. P. Braga, and S. Werner, "Set-membership affine projection algorithm for echo cancellation," *Proc. IEEE Intern. Symposium on Circuits and Systems*, Island of Kos, Greece, pp. 405-408, May 2006.

23. S. Werner, P. S. R. Diniz, and J. E. W. Moreira, "Set-membership affine projection algorithm with variable data-reuse factor," *Proc. IEEE Intern. Symposium on Circuits and Systems*, Island of Kos, Greece, pp. 261-264, May 2006.

24. L. Guo and Y.-F. Huang, "Set-membership adaptive filtering with parameter-dependent error bound tuning," *Proc. IEEE Intern. Conf. on Acoust. Speech and Signal Processing*, Philadelphia, PA, pp. IV-369 - IV-372, May 2005.

25. J. F. Galdino, J. A. Apolinário, Jr., and M. L. R. de Campos, "A set-membership NLMS algorithm with time-varying error bound," *Proc. IEEE Intern. Symposium on Circuits and Systems*, Island of Kos, Greece, pp. 277-280, May 2006.

26. R. C. de Lamare and P. S. R. Diniz, "Set-membership adaptive algorithms based on time-varying error bounds for DS-CDMA systems," *Proc. IEEE Intern. Symposium on Circuits and Systems*, Island of Kos, Greece, pp. 3466-3469, May 2006.

27. R. C. de Lamare and P. S. R. Diniz, "Blind constrained set-membership algorithms with time-varying bounds for CDMA interference suppression," *Proc. IEEE Intern. Conf. on Acoust. Speech and Signal Processing*, Toulouse, France, pp. IV-617 - IV-620, May 2006.

28. R. C. de Lamare and P. S. R. Diniz, "Set-membership adaptive algorithms based on time-varying error bounds and their application to interference suppression," *IEEE International Telecommunications Symposium*, Fortaleza, Brazil, pp. 563-567, Sept. 2006.

29. E. Hänsler and G. Schmidt, *Acoustics Echo and Noise Control: A Practical Approach*, John Wiley & Sons, Hoboken, NJ, 2004.

30. J. Benesty, T. Gänsler, D. R. Morgan, M. M. Sondhi, and S. L. Gay, *Advances in Network and Acoustics Echo Cancellation*, Springer-Verlag, Berlin, Germany, 2001.

31. International Telecommunication Union, *ITU-T G.168 Standard for Digital Network Echo Cancellers*, June 2002.

32. J. G. Proakis, *Digital Communications*, McGraw Hill, New York, NY, 4th edition, 2001.

33. M. C. Jeruchim, P. Balaban, and K. San Shanmugan, *Simulation of Communication Systems: Modeling Methodology and Techniques*, Kluwer Academic Publishers, Norwell, MA, 2nd edition, 2000.

34. European Telecommunications Standards Institute, *UMTS, technical report TR 101 112 V3.2.0*, 1998-04.

35. E. H. Dinan and B. Jabbari, "Spreading codes for direct sequence CDMA and wideband CDMA cellular networks," *IEEE Communications Magazine*, vol. 36, pp. 48-58, Sep. 1998.

36. J. I. Smith, "Computer generated multipath fading simulation for mobile radio," *IEEE Trans. on Vehicular Technology*, vol. 24, pp. 39-40, Aug. 1975.

37. T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall, Englewood Cliffs, NJ, 2nd edition, 2002.

38. P. S. R. Diniz, E. A. B. da Silva, and S. L. Netto, *Digital Signal Processing: System Analysis and Design*, Cambridge University Press, Cambridge, UK, 2002.

39. S. Werner, J. A. Apolinário Jr., M. L. R. de Campos, and P. S. R. Diniz, "Low-complexity constrained affine-projection algorithms," *IEEE Trans. on Signal Processing*, vol. 53, pp. 4545-4555, Dec. 2005.

6.12 PROBLEMS

1. In a system identification application the unknown system has transfer function given by

$$H(z) = \frac{1}{2} - \frac{1}{2}z^{-1}$$

whereas the input signal is a binary $(-1, 1)$ random signal, and the additional noise is generated via $(-\frac{1}{4}, \frac{1}{4})$ by tossing a fair coin. Evaluate by hand the first 10 iterations SM-NLMS algorithm.

2. Show that the updating equation (6.50) is equivalent to the second coefficient updating of Algorithm 6.4.

3. Repeat problem 1 for the SM-BNLMS algorithm 1.

4. Repeat problem 1 for the SM-BNLMS algorithm 2.

5. 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 consisting of a binary (-1,1) random signal. An additional Gaussian white noise with variance 10^{-2} is present at the channel output.
(a) Apply the SM-NLMS algorithm with an appropriate $\bar{\gamma}$ and find the impulse response of an equalizer with 15 coefficients.
(b) Convolve the equalizer impulse response at an iteration after convergence, with the channel impulse response and comment on the result.

6. 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 a zero-mean Gaussian noise with variance given by $\sigma_n^2 = 0.001$.

Using the SM-BNLMS algorithm:
(a) Choose an appropriate $\bar{\gamma}$, run an ensemble of 20 experiments, and plot the average learning curve.
(b) Measure the excess MSE.

7. Derive the complex versions of the SM-BNLMS algorithms 1 and 2 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) Run the algorithm for $\mu = 0.1$, $\mu = 0.4$, and $\mu = 0.8$. Comment on the convergence behavior in each case.
(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).

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

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

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

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

Algorithm 6.7

The Complex Set-Membership Affine Projection Algorithm

Initialization

$\mathbf{x}(0) = \mathbf{w}(0) = [0 \ \ldots \ 0]^T$

choose $\bar{\gamma}$ around $\sqrt{5}\sigma_n$

$\gamma =$ small constant

Do for $k \geq 0$

$\mathbf{e}^*_{\text{ap}}(k) = \mathbf{d}^*_{\text{ap}}(k) - \mathbf{X}^H_{\text{ap}}(k)\mathbf{w}(k)$

$$\mu(k) = \begin{cases} 1 - \frac{\bar{\gamma}}{|e(k)|} & \text{if } |e(k)| > \bar{\gamma} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{X}_{\text{ap}}(k)\left[\mathbf{X}^H_{\text{ap}}(k)\mathbf{X}_{\text{ap}}(k) + \gamma\mathbf{I}\right]^{-1}\mu(k)e^*(k)\mathbf{u}_1$$

The adaptive filter is also a third-order complex FIR filter, and the additional noise is zero-mean Gaussian white noise with variance $\sigma_n^2 = 0.04$. Derive and use the complex set-membership normalized LMS algorithm, choose an appropriate $\bar{\gamma}$, run an ensemble of 20 experiments, and plot the average learning curve.

9. Repeat problem 8 utilizing the complex version of SM-AP algorithm, detailed in Algorithm 6.7 provided, with $L = 4$.

10. The double threshold SM-AP algorithm can be derived for applications such as echo cancellation where there is no interest in reducing the error signal power beyond certain level. Derive an SM-AP algorithm by choosing the vector $\bar{\boldsymbol{\gamma}}(k)$ in such a way that the echo canceller does not reduce the output error power below the power of the far-end signal. Instead of using as threshold a single value of $\bar{\gamma}$, the proposed algorithm uses a range for the acceptable output error value between $\bar{\gamma}_1$ and $\bar{\gamma}_2$, where $\bar{\gamma}_1 > \bar{\gamma}_2$, as depicted in Fig. 6.31.

11. In applications where the parameters to be estimated are dominated by few dominant coefficients, that is, they are sparse, it is often desirable to employ a proportionate adaptation strategy where weights are assigned to parameter components proportional to their magnitude [20]. The updating equation of the proportionate SM-AP algorithm is given by

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{P}(k)\mathbf{X}_{\text{ap}}(k)\left[\mathbf{X}^T_{\text{ap}}(k)\mathbf{X}_{\text{ap}}(k)\right]^{-1}\left[\mathbf{e}_{\text{ap}}(k) - \bar{\boldsymbol{\gamma}}(k)\right] \tag{6.84}$$

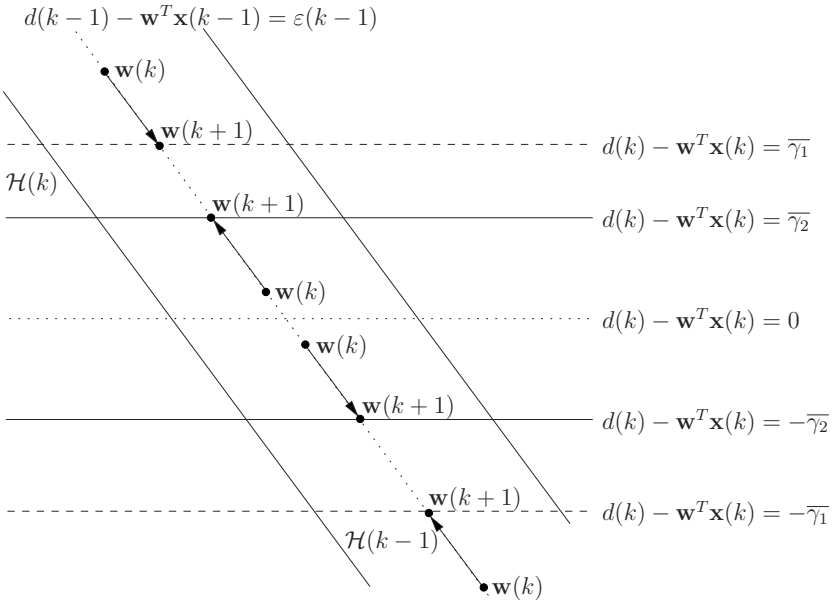


Figure 6.31 SM-AP algorithm with double threshold.

where

$$\mathbf{P}(k) = \mu(k) \begin{bmatrix} p_0(k) & 0 & \cdots & 0 \\ 0 & p_1(k) & & \vdots \\ \vdots & 0 & \cdots & \vdots \\ \vdots & \vdots & & 0 \\ 0 & 0 & \cdots & p_N(k) \end{bmatrix}$$

$$\mu(k) = \begin{cases} 1 - \frac{\bar{\gamma}}{|e(k)|} & \text{if } |e(k)| > \bar{\gamma} \\ 0 & \text{otherwise} \end{cases}$$

and

$$p_i(k) = \frac{1 - \kappa\mu(k)}{N+1} + \frac{\kappa\mu(k)|w|_i(k)}{\sum_{i=0}^{N} |w|_i(k)}$$

Use the proportionate adaption algorithm identify a system whose impulse response is given below.

$$h(k) = [1 \;\; 0 \;\; 0 \;\; 0.5 \;\; 0 \;\; 2]$$

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 $\sigma_n^2 = 5.25 \; 10^{-3}$. The adaptive filter has 6 coefficients.

(a) Use $\kappa = 0.5$, experiment some values of $\bar{\gamma}$ and discuss the results.
(b) Plot the obtained FIR filter impulse response at any iteration after convergence is achieved

and compare with the unknown system.
(c) Compare the best solution with that obtained by the corresponding SM-AP algorithm.

12. Prove from equations (6.36) to (6.39) that

$$\left\{ \left[\mathbf{x}(k)\ \tilde{\mathbf{X}}_{\mathrm{ap}}(k) \right]^T \left[\mathbf{x}(k)\ \tilde{\mathbf{X}}_{\mathrm{ap}}(k) \right] \right\} \begin{bmatrix} a \\ \mathbf{b} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

13. In SM-PUAP algorithm only $N + 1 - \bar{M}$ coefficients are updated at a given iteration. Exploit this fact to derive a reduced complexity algorithm by generalizing the procedure used to derive equations (6.36) to (6.41).

14. Identify a typical Channel A model for wireless environment described in Table 6.7 with the SM-BNLMS algorithm 2, using as input signal a Gaussian white noise and such that the signal to noise ratio at the receiver end is 10dB's. Determine though simulations the approximate number of training symbols to achieve a good channel estimation of sufficient order.