

PART I

2. Normalized LMS Algorithm with Orthogonal Correction Factors

The Normalized Least Mean Square (NLMS) algorithm [1] is a widely used adaptation algorithm due to its computational simplicity and ease of implementation. Furthermore, this algorithm is known to be robust against finite word length effects. One of the major drawbacks of the NLMS algorithm is its slow convergence for colored input signals. Over the last decade, a class of equivalent algorithms such as the Affine Projection Algorithm (APA), the Partial Rank Algorithm (PRA), and the Generalized Optimal Block Algorithm (GOBA) has been developed to ameliorate this problem [2]. These algorithms update the weights on the basis of multiple input signal vectors, while the NLMS algorithm updates the weights on the basis of a single input vector. We will refer to the entire class of algorithms as affine projection algorithms, since APA is the earliest among these algorithms and since the name APA is more widely used in the existing literature than the other names. This chapter describes a novel variant of this APA class of algorithms, which we call the NLMS algorithm with orthogonal correction factors, after the method of derivation. The approach used to derive the proposed algorithm is different from the approaches used to derive the existing algorithms. Furthermore, the input vectors used to adapt the weights are chosen differently. Our choice of input vectors leads, under most circumstances, to faster convergence than the choice proposed in earlier works [2, 19, 23].

2.1 NLMS Algorithm

Figure 2.1 shows an adaptive filter used in the system identification mode. Here, the system input x_n and corresponding measured output d_n , possibly after contamination with measurement noise ε_n , are known. The objective is to estimate an N dimensional weight vector $\hat{\mathbf{w}}_n$, such that the estimated output $\hat{d}_n = \hat{\mathbf{w}}_n^H \mathbf{x}_n$, where $\mathbf{x}_n = (x_n, x_{n-1}, \dots, x_{n-N+1})^t$ is the input vector at the n th instant, is as close as possible to the measured output d_n in mean-squared error sense. The NLMS algorithm is an iterative procedure to estimate these weights.

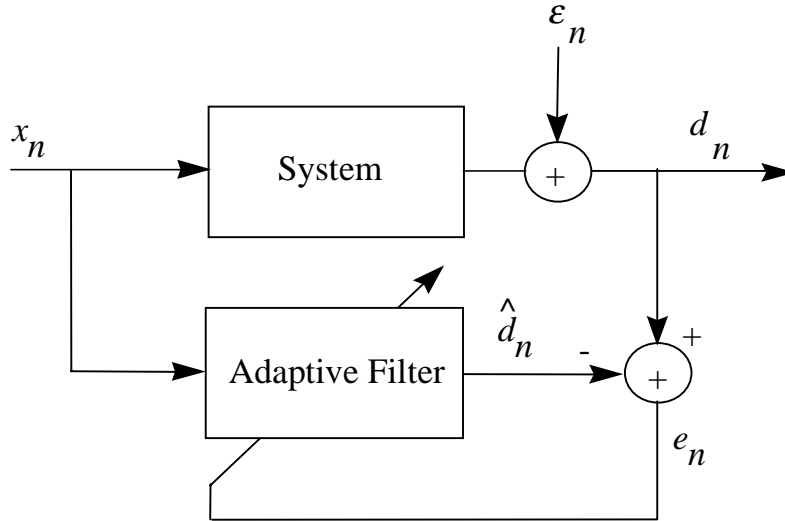


Figure 2.1 Adaptive Filtering Problem.

The following steps constitute the NLMS algorithm [1]:

1. *Filter output:*

$$\hat{d}_n = \hat{\mathbf{w}}_n^H \mathbf{x}_n \quad (2.1)$$

2. *Estimation error:*

$$e_n = d_n - \hat{d}_n \quad (2.2)$$

3. *Tap weight adaptation:*

$$\hat{\mathbf{w}}_{n+1} = \hat{\mathbf{w}}_n + \frac{\bar{\mu} \mathbf{x}_n e_n^*}{\|\mathbf{x}_n\|^2} \quad (2.3)$$

The variable $\bar{\mu}$ is known as the step-size. For $\bar{\mu} = 1$, (2.3) adapts the tap weights such that the *a posteriori* estimation error is zero. That is,

$$\hat{\mathbf{w}}_{n+1}^H \mathbf{x}_n = d_n. \quad (2.4)$$

If the desired output $d_n = \mathbf{w}^{0H} \mathbf{x}_n$ comes from a finite impulse response (FIR) system, with weights \mathbf{w}^0 that can be captured by the model (in other words, the model has a sufficient order),

and there is no measurement noise, the minimum achievable mean-square estimation error is zero. Under this condition, for $\bar{\mu} = 1$, the error in the weight estimate $\tilde{\mathbf{w}}_n = \mathbf{w}^0 - \hat{\mathbf{w}}_n$ and the new estimate for the weights $\hat{\mathbf{w}}_{n+1}$ are related as follows:

$$\begin{aligned} \tilde{\mathbf{w}}_{n+1}^H [\hat{\mathbf{w}}_{n+1} - \hat{\mathbf{w}}_n] &= \frac{e_n^* \tilde{\mathbf{w}}_{n+1}^H \mathbf{x}_n}{\|\mathbf{x}_n\|^2} \\ &= \frac{e_n^* (\mathbf{w}^{0H} \mathbf{x}_n - \hat{\mathbf{w}}_{n+1}^H \mathbf{x}_n)}{\|\mathbf{x}_n\|^2} \\ &= 0 \end{aligned} \quad (2.5)$$

A geometric interpretation of (2.5) is that $\hat{\mathbf{w}}_{n+1}$, the new estimate for the weights, is the point that is nearest (in L_2 sense) to the true weights \mathbf{w}^0 , along the direction specified by the input vector \mathbf{x}_n . This is illustrated in Figure 2.2.

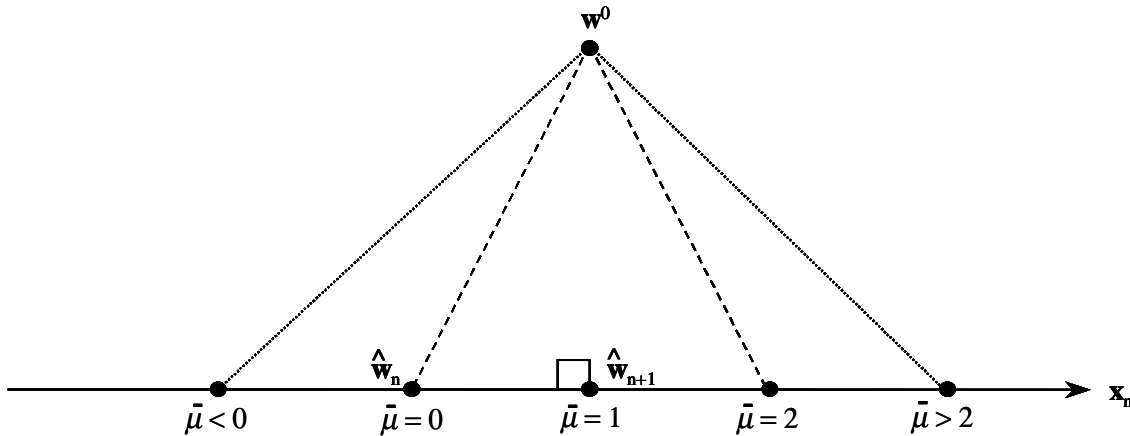


Figure 2.2 Estimates Produced by NLMS for Various Values of $\bar{\mu}$.

From this illustration, it is also evident that if $\bar{\mu} \notin [0,2]$ then the distance from the estimated weights to the true weights increases during successive iterations, and hence the weights diverge.

If N consecutive input vectors \mathbf{x}_n are orthogonal, the NLMS algorithm, starting from any arbitrary weights $\hat{\mathbf{w}}_0$, minimizes the distance from the true weights \mathbf{w}^0 along N orthogonal directions and thereby converges to the true weights in exactly N iterations. In practice, the consecutive input vectors are rarely orthogonal. In particular, when the input signal is strongly colored, the successive input vectors tend to be almost parallel to each other. The weight estimates then improve very little during successive iterations, as shown in Figure 2.3.

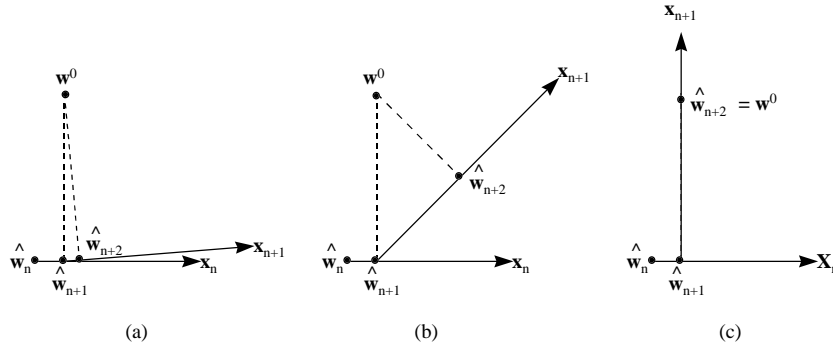


Figure 2.3 Successive Estimates Produced by NLMS (a) Almost Parallel Input - Small Improvement, (b) Differently Oriented Input - Large Improvement, (c) Orthogonal Input - Most Improvement.

Furthermore, when the signal is colored, the input vectors are not equally likely to be oriented in all directions. They tend to span the major eigenspaces, thereby slowing convergence in the direction of the minor eigenspaces. The fact that most of the signal energy is concentrated in the major eigenspace can be proved analytically using the Karhunen-Loève expansion [1].

Figure 2.4 shows the scatter plot of the input vectors of a low-pass signal having $0.2f_s$ as the corner frequency. We see that the input vectors are oriented more frequently along the first- and third-quadrants, and less frequently along the second- and fourth-quadrants. This behavior motivates us to introduce orthogonal correction factors to NLMS to accelerate its convergence, as will be explained in Section 2.3.

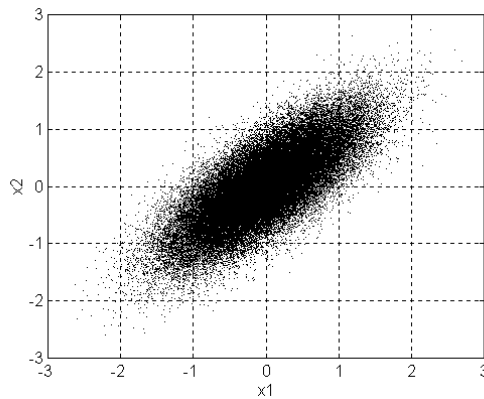


Figure 2.4 Scatter Plot of the Input Vector Corresponding to a Low-Pass Signal.

2.2 NLMS with Orthogonal Correction Factors

As depicted in Figure 3, the best improvement in weights occurs if the successive input vectors (corrections) are orthogonal. When the successive input vectors are not orthogonal, we propose generating appropriate orthogonal directions and moving suitably along those directions. This procedure is similar to the method of Rosenbrock [16] that minimizes nonlinear functions by successively minimizing along orthogonal directions. Thus, the weight adaptation equation of the NLMS algorithm is modified as follows:

$$\hat{\mathbf{w}}_{n+1} = \hat{\mathbf{w}}_n + \mu_0 \mathbf{x}_n + \mu_1 \mathbf{x}_n^1 + \dots + \mu_M \mathbf{x}_n^M, \quad n \geq M < N \quad (2.6)$$

where the vectors $\mathbf{x}_n, \mathbf{x}_n^1, \mathbf{x}_n^2, \dots, \mathbf{x}_n^M$ are orthogonal to each other. If $M = N - 1$, under noise-free conditions, exact convergence is possible, i.e. $\hat{\mathbf{w}}_{n+1} = \mathbf{w}^0$. The procedure to generate the orthogonal directions and corresponding step-lengths is explained below.

As before, \mathbf{x}_n is the input vector at the n th instant, and μ_0 is chosen as in NLMS.

$$\mu_0 = \frac{\bar{\mu} e_n^*}{\|\mathbf{x}_n\|^2} \quad (2.7)$$

Let $\hat{\mathbf{w}}_{n+1}^1 = \hat{\mathbf{w}}_n + \mu_0 \mathbf{x}_n$ be the new estimate for the weights, obtained after the correction along \mathbf{x}_n . Using the Gram-Schmidt procedure [17], we write \mathbf{x}_{n-D} ($D > 0$) as the sum of a component along \mathbf{x}_n and a component orthogonal to \mathbf{x}_n . Let the orthogonal component be \mathbf{x}_n^1 . Then, the step-size along \mathbf{x}_n^1 is chosen as

$$\mu_1 = \begin{cases} \frac{\bar{\mu} e_n^{1*}}{\|\mathbf{x}_n^1\|^2} & \text{if } \|\mathbf{x}_n^1\| \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad (2.8)$$

where $e_n^1 = d_{n-D} - \hat{\mathbf{w}}_{n+1}^{1H} \mathbf{x}_{n-D}$ is the error in estimating d_{n-D} using $\hat{\mathbf{w}}_{n+1}^1$. For $\bar{\mu} = 1$, the above choice of μ_1 minimizes the a posteriori estimation error at $n - D$, with $\hat{\mathbf{w}}_{n+1}^1$ as weights.

Generalizing the above steps, \mathbf{x}_n^k is the component of \mathbf{x}_{n-kD} that is orthogonal to $\mathbf{x}_n, \mathbf{x}_{n-D}, \mathbf{x}_{n-2D}, \dots, \mathbf{x}_{n-(k-1)D}$ and it can be computed using the Gram-Schmidt procedure [17]. The corresponding step size μ_k is calculated according to

$$\mu_k = \begin{cases} \frac{\bar{\mu} e_n^{k*}}{\|\mathbf{x}_n^k\|^2} & \text{if } \|\mathbf{x}_n^k\| \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad (2.9)$$

where

$$e_n^k = d_{n-kD} - \hat{\mathbf{w}}_{n+1}^{kH} \mathbf{x}_{n-kD}, \text{ and}$$

$$\hat{\mathbf{w}}_{n+1}^k = \hat{\mathbf{w}}_n + \mu_0 \mathbf{x}_n + \mu_1 \mathbf{x}_n^1 + \dots + \mu_{k-1} \mathbf{x}_n^{k-1}.$$

For $\bar{\mu} = 1$, the above choice of μ_k minimizes the *a posteriori* estimation error at $n - kD$, with $\hat{\mathbf{w}}_{n+1}^{k+1}$ as weights. It is worthwhile to point out that the above orthogonalization based procedure, with $\bar{\mu} = 1$, can also be used to obtain the minimum norm solution of an under-determined system of linear equations.

Table 2.1 summarizes the NLMS-OCF algorithm, along with the number of computations needed in each step. Adding the number of computations needed in each step, the total number of computations needed per NLMS-OCF iteration is

$$2N + 4 + \sum_{k=1}^M [3N + 2 + (2N + 1)k] = NM^2 + 4NM + M^2/2 + 2N + 5M/2 + 4 \quad (2.10)$$

Summarizing (2.10), the NLMS-OCF algorithm has a computational complexity of $O(NM^2)$.

Table 2.1 Summary of the NLMS-OCF algorithm.

Choose an arbitrary $\hat{\mathbf{w}}_0$. Choose the number of Orthogonal Correction Factors $M < N$. Pick $\bar{\mu} \in (0,2)$. Repeat the following steps for each n.	Number of Computations
(1) $e_n = d_n - \mathbf{x}_n^T \hat{\mathbf{w}}_n$	N
(2) $\ \mathbf{x}_n\ ^2 = \ \mathbf{x}_{n-1}\ ^2 + x_n^2 - x_{n-N}^2$	2
(3) $\mu_0 = \frac{\bar{\mu} e_n}{\ \mathbf{x}_n\ ^2}$	2
(4) $\hat{\mathbf{w}}_{n+1}^1 = \hat{\mathbf{w}}_n + \mu_0 \mathbf{x}_n$	N
(5) $\mathbf{x}_n^0 = \mathbf{x}_n$	
For $k = 1, 2, \dots, M$, repeat steps (6)-(9)	
(6) $\mathbf{x}_n^k = \mathbf{x}_{n-kD} - \sum_{i=0}^{k-1} \frac{\mathbf{x}_{n-kD}^T \mathbf{x}_n^i}{\ \mathbf{x}_n^i\ ^2} \mathbf{x}_n^i$	$(2N + 1)k$
(7) $e_n^k = d_{n-kD} - \mathbf{x}_{n-kD}^T \hat{\mathbf{w}}_{n+1}^k$	N
(8) $\mu_k = \begin{cases} \frac{\bar{\mu} e_n^k}{\ \mathbf{x}_n^k\ ^2} & \text{if } \ \mathbf{x}_n^k\ \neq 0 \\ 0 & \text{otherwise} \end{cases}$	$N + 2$
(9) $\hat{\mathbf{w}}_{n+1}^{k+1} = \hat{\mathbf{w}}_{n+1}^k + \mu_k \mathbf{x}_n^k$	N
(10) $\hat{\mathbf{w}}_{n+1} = \hat{\mathbf{w}}_{n+1}^{M+1}$	

2.3 Equivalence between NLMS-OCF and APA

The affine projection algorithm is a special case of NLMS-OCF where the delay D is set to unity as explained in this section. When $\bar{\mu} = 1$, the weight update generated by APA is the vector that is as close as possible to the current weight vector, while setting the most recent $(M + 1)$ *a posteriori* error estimates to zero [3]. That is,

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \Delta \mathbf{w}_n \quad (2.11)$$

where $\Delta \mathbf{w}_n$ is the minimum-norm solution to

$$\mathbf{X}_n^T \Delta \mathbf{w}_n = \mathbf{e}_n \quad (2.12)$$

In the above equation, $\mathbf{X}_n = [\mathbf{x}_n \quad \mathbf{x}_{n-1} \quad \cdots \quad \mathbf{x}_{n-M}]$, $\mathbf{e}_n = [e_n \quad \tilde{e}_n^1 \quad \cdots \quad \tilde{e}_n^M]^T$, $e_n = d_n - \mathbf{x}_n^T \mathbf{w}_n$, and $\tilde{e}_n^k = d_{n-k} - \mathbf{x}_{n-k}^T \mathbf{w}_n$. Since $\Delta \mathbf{w}_n$ is the minimum-norm solution of (2.12), it is the unique solution of (2.12) that lies in the space spanned by the columns of \mathbf{X}_n . APA usually solves for $\Delta \mathbf{w}_n$ using the following matrix equation.

$$\Delta \mathbf{w}_n = \mathbf{X}_n [\mathbf{X}_n^T \mathbf{X}_n]^{-1} \mathbf{e}_n \quad (2.13)$$

Observe that the above solution lies in the space spanned by the columns of \mathbf{X}_n . Simple algebra shows that \mathbf{w}_{n+1} obtained using (2.11) and (2.13) sets the most recent $(M + 1)$ *a posteriori* error estimates to zero. That is,

$$\mathbf{X}_n^T \mathbf{w}_{n+1} = [d_n \quad d_{n-1} \quad \cdots \quad d_{n-M}]^T. \quad (2.14)$$

NLMS-OCF, on the other hand, finds the weight update by setting "one *a posteriori* estimation error at a time to zero," as explained below. NLMS-OCF begins by setting the *a posteriori* estimation error at n to zero, while keeping the norm of the increment in weights to a minimum. That is, it finds the weight \mathbf{w}_n^1 such that $\|\mathbf{w}_n^1 - \mathbf{w}_n\|$ is minimized subject to $d_n - \mathbf{x}_n^T \mathbf{w}_n^1 = 0$. This solution is given by,

$$\mathbf{w}_n^1 = \mathbf{w}_n + \mu_0 \mathbf{x}_n \quad (2.15)$$

where $\mu_0 = \frac{e_n}{\mathbf{x}_n^T \mathbf{x}_n}$ and $e_n = d_n - \mathbf{x}_n^T \mathbf{w}_n$.

Next, NLMS-OCF finds the weight \mathbf{w}_n^2 that forces the *a posteriori* estimation error at $(n-1)$ to zero, while maintaining the zero *a posteriori* estimation error at n , and keeping the norm of the increment in weights to a minimum. That is, find the weight \mathbf{w}_n^2 such that $\|\mathbf{w}_n^2 - \mathbf{w}_n\|$ is minimized subject to $d_n - \mathbf{x}_n^T \mathbf{w}_n^2 = 0$ and $d_{n-1} - \mathbf{x}_{n-1}^T \mathbf{w}_n^2 = 0$. If the increment in weights $(\mathbf{w}_n^2 - \mathbf{w}_n^1)$ is orthogonal to \mathbf{x}_n , then $d_n - \mathbf{x}_n^T \mathbf{w}_n^2 = d_n - \mathbf{x}_n^T \mathbf{w}_n^1 = 0$. Thus the first constraint is satisfied if the weight increment is orthogonal to \mathbf{x}_n . Hence, we decompose \mathbf{x}_{n-1} into a component along \mathbf{x}_n and a component \mathbf{x}_n^1 that is orthogonal to \mathbf{x}_n . We increment the weights along \mathbf{x}_n^1 such that the second constraint is satisfied. This solution is given by,

$$\begin{aligned}\mathbf{w}_n^2 &= \mathbf{w}_n^1 + \mu_1 \mathbf{x}_n^1 \\ &= \mathbf{w}_n + \mu_0 \mathbf{x}_n + \mu_1 \mathbf{x}_n^1\end{aligned}\tag{2.16}$$

where $\mu_0 = \frac{e_n^1}{\mathbf{x}_n^1 T \mathbf{x}_n^1}$ and $e_n^1 = d_{n-1} - \mathbf{x}_{n-1}^T \mathbf{w}_n^1$.

The above process is repeated until each of the most recent $(M+1)$ *a posteriori* errors is forced to zero. We describe here the general step that forces the *a posteriori* estimation error at $(n-k)$ to zero, where $k \in \{1, 2, \dots, M\}$. Here, we find the weight \mathbf{w}_n^k such that $\|\mathbf{w}_n^k - \mathbf{w}_n\|$ is minimized subject to $d_{n-r} - \mathbf{x}_{n-r}^T \mathbf{w}_n^k = 0$ for $r = 0, 1, \dots, k-1$, and $d_{n-k} - \mathbf{x}_{n-k}^T \mathbf{w}_n^k = 0$. If the increment in weights, $(\mathbf{w}_n^k - \mathbf{w}_n^{k-1})$, is orthogonal to $\mathbf{x}_n, \mathbf{x}_{n-1}, \dots, \mathbf{x}_{n-(k-1)}$, then $d_{n-r} - \mathbf{x}_{n-r}^T \mathbf{w}_n^k = d_{n-r} - \mathbf{x}_{n-r}^T \mathbf{w}_n^{k-1} = 0$ for $r = 0, 1, \dots, k-1$. Thus, the first k constraints are satisfied if the increment is orthogonal to $\mathbf{x}_n, \mathbf{x}_{n-1}, \dots, \mathbf{x}_{n-(k-1)}$. Hence, we decompose \mathbf{x}_{n-k} into a component that is in the span of $\mathbf{x}_n, \mathbf{x}_{n-1}, \dots, \mathbf{x}_{n-(k-1)}$ and a component \mathbf{x}_n^k that is orthogonal to $\mathbf{x}_n, \mathbf{x}_{n-1}, \dots, \mathbf{x}_{n-(k-1)}$. We increment the weights along \mathbf{x}_n^k such that the last constraint is satisfied. This solution is given by,

$$\begin{aligned}\mathbf{w}_n^{k+1} &= \mathbf{w}_n^k + \mu_1 \mathbf{x}_n^k \\ &= \mathbf{w}_n + \mu_0 \mathbf{x}_n + \mu_1 \mathbf{x}_n^1 + \dots + \mu_k \mathbf{x}_n^k\end{aligned}\tag{2.17}$$

where $\mu_k = \frac{e_n^k}{\mathbf{x}_n^{kT} \mathbf{x}_n^k}$ and $e_n^k = d_{n-k} - \mathbf{x}_{n-k}^T \mathbf{w}_n^k$.

Thus the weight update that forces the most recent $(M + 1)$ *a posteriori* estimation errors to zero is given by

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu_0 \mathbf{x}_n + \mu_1 \mathbf{x}_n^1 + \dots + \mu_M \mathbf{x}_n^M \quad (2.18)$$

where $M + 1$ is the number of input vectors used for adaptation, \mathbf{x}_n is the input vector at the n th instant, \mathbf{x}_n^k , for $k = 1, 2, \dots, M$, is the component of \mathbf{x}_{n-k} that is orthogonal to $\mathbf{x}_n, \mathbf{x}_{n-1}, \mathbf{x}_{n-2}, \dots, \mathbf{x}_{n-(k-1)}$, and μ_k , for $k = 0, 1, \dots, M$ is chosen as in (2.19).

$$\mu_k = \begin{cases} \frac{e_n}{\mathbf{x}_n^T \mathbf{x}_n} & \text{for } k = 0, \text{ if } \|\mathbf{x}_n\| \neq 0 \\ \frac{e_n^k}{\mathbf{x}_n^{kT} \mathbf{x}_n^k} & \text{for } k = 1, 2, \dots, M, \text{ if } \|\mathbf{x}_n^k\| \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad (2.19)$$

where

$$\begin{aligned} e_n &= d_n - \mathbf{x}_n^T \mathbf{w}_n, \\ e_n^k &= d_{n-k} - \mathbf{x}_{n-k}^T \mathbf{w}_n^k, \text{ for } k = 1, 2, \dots, M, \text{ and} \\ \mathbf{w}_n^k &= \mathbf{w}_n + \mu_0 \mathbf{x}_n + \mu_1 \mathbf{x}_n^1 + \dots + \mu_{k-1} \mathbf{x}_n^{k-1}. \end{aligned} \quad (2.20)$$

Observe from (2.18) that the increment in weight lies in the space spanned by the columns of \mathbf{X}_n . Furthermore, the updated weight satisfies (2.14). Equivalently, the weight increment satisfies (2.12). Since the minimum-norm solution to (2.12) is the unique solution of (2.12) that is in the space spanned by the columns of \mathbf{X}_n , the weight updates generated by APA and by NLMS-OCF with $D = 1$ are identical.

As is usually done in APA, the above algorithm can be generalized by introducing a constant $\bar{\mu}$, usually referred to as the step size. This generalization, along with the modifications needed for the complex case, results in the update equations (2.6) and (2.9).

2.4 Simulation Results

Figure 2.5 shows the locus of weight estimates $\hat{\mathbf{w}}_n$ generated by NLMS and NLMS with one orthogonal correction factor, termed NLMS-OCF(1).

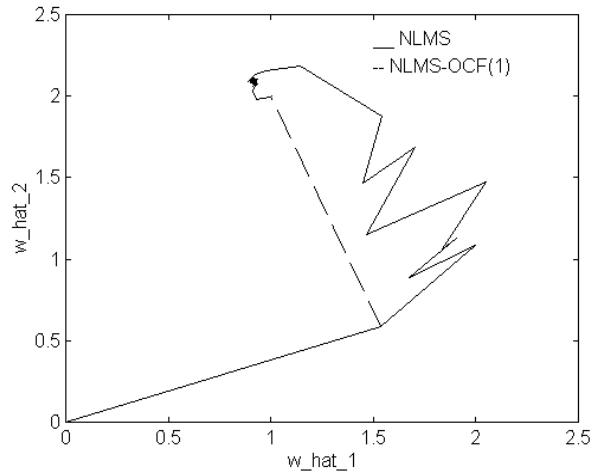


Figure 2.5 Locus of Weight Estimates Generated by NLMS and NLMS-OCF(1).

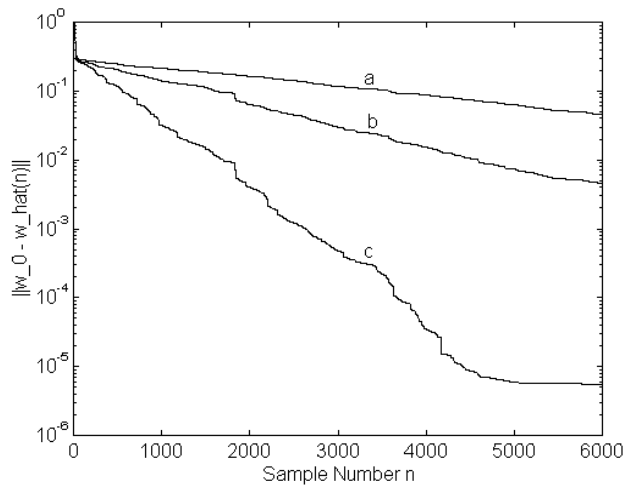
The desired signal is generated by a first order FIR filter with impulse response $\{1,2\}$. The input to the system is a low pass signal with $0.01f_s$ as the corner frequency and the step-size $\bar{\mu}$ is 1. While the NLMS algorithm goes through a lot of zigzagging before reaching the true weights, NLMS-OCF with $M (= N - 1) = 1$ estimates the true weights in two steps.

Figure 2.6 shows the performance of the NLMS and NLMS-OCF algorithms for different values of M and for different input signals. The lowpass, highpass, and bandpass signals have dynamic ranges, indicated by their max-to-min eigenvalue ratios, of approximately 20,000, 10,000, and 100 respectively. Here, the true system has $\{0.9821, -0.0092, -0.0019, -0.0157, 0.0179, 0.0248, 0.0038, 0.0246\}$ as its impulse response.

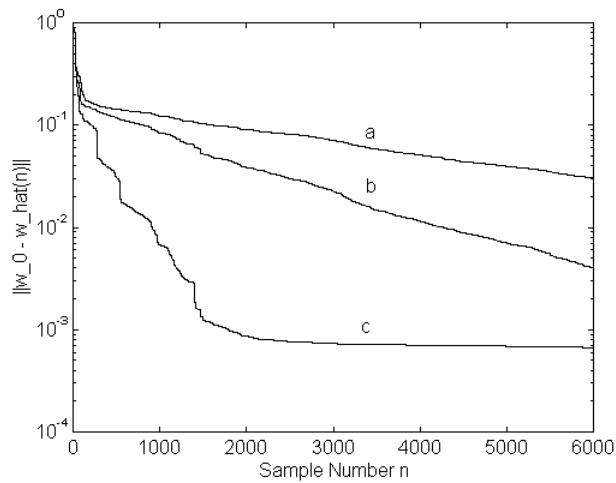
We observe that increasing the number of orthogonal correction factors M improves the convergence rate in each case. We also see that the convergence rate is different for same-bandwidth signals in different spectral regions, and that this roughly corresponds to their respective dynamic ranges. Extrapolating all three cases in Figure 6, we note that in each case NLMS will reach the final performance level (indicated by the level at which NLMS-OCF(2) levels out) at a number of iterations equal to about 10 times that for NLMS-OCF(2). For these

examples, at the cost of computing 2 orthogonal correction factors, the convergence rate has improved by an order of magnitude.

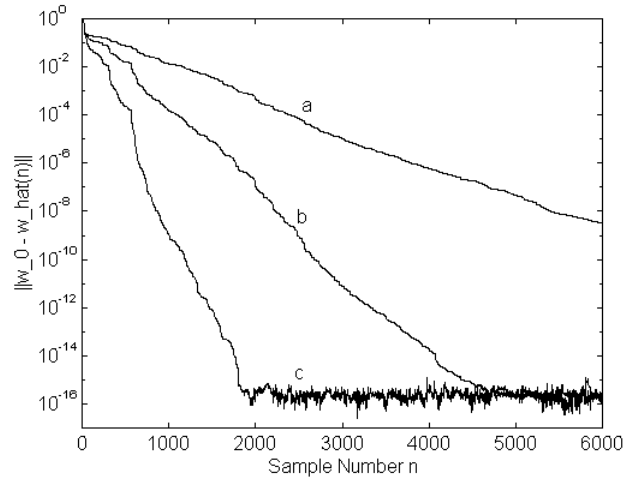
While correction factors increase the convergence rate, at some point this becomes computationally too expensive. For the present example, with 2 orthogonal correction factors, the added computational cost is still less than that necessary to keep running NLMS (for more iterations) until the same performance is achieved. For some applications it will be important that NLMS-OCF requires fewer iterations, and therefore fewer data, to achieve the same performance.



(i)



(ii)



(iii)

Figure 2.6 Results from (a) NLMS, (b) NLMS-OCF(1), and (c) NLMS-OCF(2) with (i) Low-Pass ($0, 0.25f_s$), (ii) High-Pass ($0.25f_s, 0.5f_s$), and (iii) Band-Pass ($0.125f_s, 0.375f_s$) Inputs.

2.5 Conclusion

The NLMS algorithm converges slowly, especially when the input signal is colored. Using orthogonal correction factors mitigates this problem. A procedure to determine the orthogonal correction factors is presented. Simulation results indicate that the convergence rate improves as the number of orthogonal correction factors increases. The number of orthogonal correction factors can be chosen based on the required convergence rate and the available computational power. Under noise-free conditions, the NLMS-OCF algorithm converges in a finite number of iterations if the number of correction factors equals the system order, i.e. $M = N - 1$. The only disadvantage of the NLMS-OCF algorithm, in the form presented above, is its higher complexity, especially for M greater than 2 or 3. We will develop a so-called fast version of NLMS-OCF in Chapter 4.