Contraction Mapping: An Important Property in Adaptive Filters

MARKUS RUPP*

Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA 93106-9560
e-mail: markus@agni.ece.ucsb.edu

Abstract– In this paper we show that many adaptive filters used for system identification are contraction mappings. Applying deterministic methods we give conditions under which algorithms, like Least Mean Square, Normalized Least Mean Square, Modified Least Mean Square with Delayed update, Modified Filtered-X Least Mean Square, Affine Projection, and Recursive Least Square are a contraction mapping contracting. Based on this result, we investigate the algorithms' convergence rate for initialization phase and tracking.

I. INTRODUCTION

Although adaptive filter have been used successfully for system identification during the last three decades a unique framework addressing all important properties does not exist. Even excellent textbooks like [4] use stochastic approaches for the derivation of the Least Mean Square algorithm (LMS) and deterministic approaches for the derivation of the Recursive Least Square (RLS) algorithm. Recently, following a determistic approach, it has been shown that the LMS algorithm is H^∞ optimal [9]. However, other gradient-type algorithms have not yet been included. In this paper we avoid the necessity of minimizing a performance index and follow the idea that an estimate of an unknown plant should be improved with every update step of a given algorithm. Such a behavior can be described by contraction mapping (CM), which is a well-known property in the field of solving (non)linear equations and convergence theory of sequences. Let us first briefly describe the identification problem as described in Fig. 1. An unknown plant $\underline{w}(k)$ is driven by a sequence $u(k)$. The observation $d(k)$ at the output is corrupted by additive noise $v(k)$. Both observations, i.e., input $u(k)$ and output $d(k)$, are filtered by $A(k)$ and $B(k)$, respectively. With the filtered outputs $\underline{\psi}(k), \underline{\phi}(k)$ and $d_\phi(k)$ the algorithm estimates the unknown plant resulting in an $(1 \times M)$ estimation vector $\underline{\hat{w}}(k)$. Instead of the (output) error $e(k)$, the (filtered) error $e_\phi(k)$ is used to update the coefficients. Thus, a general formulation for the coefficient update and the error equation is

$$\underline{\hat{w}}(k+1) = \underline{\hat{w}}(k) + \mu(k)\,\underline{\psi}(k) e_\phi(k) \tag{1}$$

$$e_\phi(k) = d_\phi(k) - \underline{\hat{w}}^T(k)\underline{\phi}(k)\,, \tag{2}$$

where $\mu(k)$ is a step-size parameter. As depicted in Fig. 1,

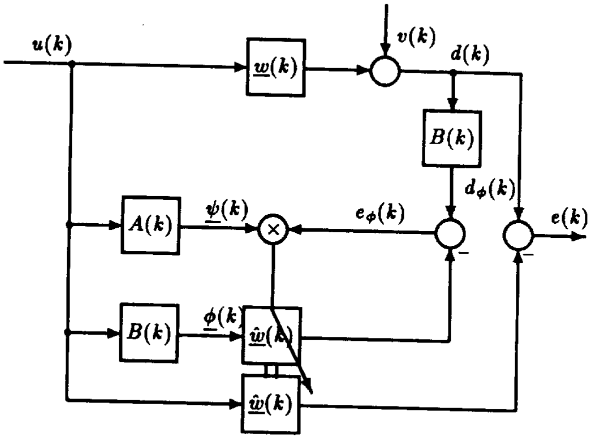

Figure 1: Adaptive filter structure for system identification

two linear FIR filters A, B can be applied to the input vector.

$$\underline{\psi}(k) = \sum_{i=0}^{F_a} a_i(k)\underline{u}(k-i) = A(k)[\underline{u}(k)] \tag{3}$$

$$\underline{\phi}(k) = \sum_{i=0}^{F_b} b_i(k)\underline{u}(k-i) = B(k)[\underline{u}(k)]\,. \tag{4}$$

This special choice of the vectors $\underline{\psi}(k), \underline{\phi}(k)$ allows us to describe all common algorithms in a unique framework. Since the output $d(k)$ of the system can be constructed as a linear combination of the input values plus additive noise $v(k)$,

$$d(k) = \underline{w}^T(k)\underline{u}(k) + v(k)\,, \tag{5}$$

$$d_\phi(k) = \sum_{i=0}^{F_b} b_i(k) d(k-i)\,, \tag{6}$$

the error equation (2) can be rewritten to

$$\begin{aligned} e_\phi(k) &= d_\phi(k) - \underline{\hat{w}}^T(k)\underline{\phi}(k) && (7)\\ &\simeq \underline{w}^T(k)\underline{\phi}(k) - \underline{\hat{w}}^T(k)\underline{\phi}(k) + v_\phi(k) && (8)\\ &= \underline{\epsilon}^T(k)\underline{\phi}(k) + v_\phi(k)\,, && (9)\end{aligned}$$

where we have introduced the weight-error vector $\underline{\epsilon}(k) = \underline{w}(k) - \underline{\hat{w}}(k)$ and the filtered noise $v_\phi(k) = B(k)[v(k)]$. In

*This work was supported by a scholarship from DAAD (German Academic Exchange Service) and the scientific division of NATO (Wissenschaftsausschuß der NATO).

0-7803-1948-6/94 $4.00 © 1994 IEEE

(8) we assumed that the system changes only very slowly. This condition is necessary when filtered desired values $d_{\phi}(k)$ are used. Many algorithm, however, use $B = I$, thus they avoid the filtering at all and (8) is exact. An additional effect arises in the noise term $v_{\phi}(k)$, since it is now filtered as well. To apply the concept of CM, it is necessary to write the update equation in state-space form, which can be done by substituting the weight-error vector in (1)

$$\underline{\epsilon}(k+1) = \left(\underline{I} - \mu(k)\underline{\psi}(k)\underline{\phi}^T(k)\right)\underline{\epsilon}(k) - \mu(k)\underline{\psi}(k)v_{\phi}(k) \quad (10)$$

where $\underline{I}$ is an M dimensional identity matrix. For achieving CM, the inhomogeneous part of the state-space formulation has to be neglected. Thus, all algorithms in this paper will be regarded as noiseless. Although, this is a very restrictive simplification, most properties can be investigated correctly.

II. Contraction mapping

Definition 1 *If an operator T is defined on a domain D in a normed linear space, and if there is a constant $0 < C < 1$ such that*

$$\|Tx_1 - Tx_2\| \leq C\|x_2 - x_1\|$$

then T is a contraction operator (contraction mapping).

Comparing (10) with the above definition we can use the matrix $\left(\underline{I} - \mu(k)\underline{\psi}(k)\underline{\phi}^T(k)\right)$ as the operator $\underline{T}$. In this paper we will focus on the properties of $\underline{T}$ and find vectors $\underline{\psi}(k), \underline{\phi}(k)$ such that the matrix is a CM. The squared L_2-norm of the weight-error vector leads to convenient expressions

$$\begin{aligned} \|\underline{\epsilon}(k+1)\|_2^2 &= \left\|\left(\underline{I} - \mu(k)\underline{\psi}(k)\underline{\phi}^T(k)\right)\underline{\epsilon}(k)\right\|_2^2 & (11) \\ &= \|\underline{\epsilon}(k)\|_2^2 \; C(k)\,, & (12) \end{aligned}$$

where $C(k)$ equals

$$1 - \frac{2\mu(k)\underline{\psi}^T(k)\underline{\epsilon}(k)\underline{\phi}^T(k)\underline{\epsilon}(k) - \mu^2(k)\|\underline{\psi}(k)\|_2^2\left(\underline{\phi}^T(k)\underline{\epsilon}(k)\right)^2}{\|\underline{\epsilon}(k)\|_2^2}\,.$$

Thus, we have to find conditions such that $0 < C(k) < 1$. Obviously, not every vector $\underline{\psi}(k), \underline{\phi}(k)$ leads to CM; for example vectors that are orthogonal to $\underline{\epsilon}(k)$ cannot lead to an improvement and must be excluded from the set of allowed vectors. Fig. 2 depicts the situation graphically. Given a weight-error vector $\underline{\epsilon}(k)$ and an update direction $\underline{\psi}(k)$ what amount of $\underline{\psi}(k)$ has to be added in order to obtain CM? Applying some geometry, the figure leads to the following basic theorem.

Theorem 1 *a) For any given direction $\underline{\psi}(k)$ not orthogonal to $\underline{\epsilon}(k)$ a certain amount of $\underline{\psi}(k)$ can be added to $\underline{\epsilon}(k)$ such that $\underline{\epsilon}(k+1)$ will become smaller.*

b) The maximal range $\Delta\underline{\epsilon}(k)$ that contracts $\underline{\epsilon}(k)$ is given by

$$\Delta\underline{\epsilon}(k) = 2\frac{\underline{\psi}^T(k)\underline{\epsilon}(k)}{\underline{\psi}^T(k)\underline{\psi}(k)}\underline{\psi}(k)\,.$$

c) If a certain amount $\Delta\underline{\epsilon}(k)$ is known to cause a contraction than every amount $\alpha\Delta\underline{\epsilon}(k)$ causes a contraction for $0 < \alpha \leq 1$.

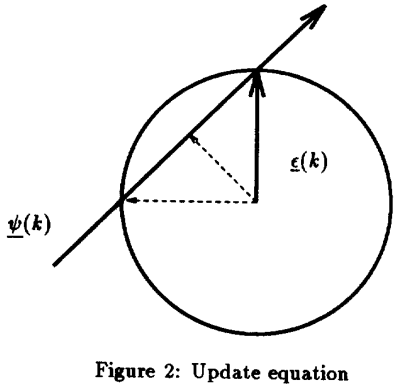

Figure 2: Update equation

We can separate the algorithms into two classes: symmetric and unsymmetric. Symmetric algorithms use only one vector $\underline{\psi}(k) = \underline{\phi}(k)$, whereas unsymmetric algorithms use two different vectors.

A. Symmetric Algorithms

In the case of a symmetric algorithm ($A(k) = B(k)$) CM is obtained for positive step-sizes

$$0 < \mu(k) = \frac{\alpha}{\|\underline{\psi}(k)\|_2^2} \quad ;\text{for } 0 < \alpha < 2\,, \qquad (13)$$

which is in agreement with Theorem 1b. It can be expected that the convergence speed can be maximized by finding a specific α. Derivating with respect to α, the solution for maximal convergence speed reads $\alpha_{os} = 1$. This solution, however, implies that the choice of α does not influence the term $\kappa_{\psi}^2(k) = \left|\underline{\psi}^T(k)\underline{\epsilon}(k)\right|^2 / (\|\underline{\psi}(k)\|_2^2\|\underline{\epsilon}(k)\|_2^2$. The resulting value of $C(k)$ is $C_{\min}(k) = 1 - \kappa_{\psi}^2(k)$. Thus, the convergence speed is still dependent on $\kappa_{\psi}^2(k)$. Table 1 depicts a list of algorithms that fall into the category of symmetric algorithms.

$\underline{\psi}(k) = \underline{u}(k)$	LMS algorithm
$\underline{\psi}(k) = \underline{u}(k-D)$	Modified DLMS alg. [8]
$\underline{\psi}(k) = \sum_{i=0}^{F_a} a_i\underline{u}(k-i)$	Modified FXLMS alg. [2]
$\underline{\psi}(k) = \sum_{i=0}^{F_a} a_i(k)\underline{u}(k-i)$	$A(k)$ for decorrelation [10]

Table 1: Special cases for symmetric algorithms

B. Unsymmetric Algorithms

If we use $B(k) \neq A(k)$ we obtain an unsymmetric algorithm. Since a filter $B(k)$ always causes the drawback of permitting only a slowly time-varying system, common algorithms use the instantaneous error $e(k)$ instead of a filtered one ($B = I$, or equivalently $\underline{\phi}(k) = \underline{u}(k)$) resulting in

$$\underline{T}(k) = \underline{I} - \mu(k)\underline{\psi}(k)\underline{u}^T(k)\,. \qquad (14)$$

In this case several well-known algorithms fall into the unsymmetric category: Block Least Square (BLS), Ozeki/Umeda's Affine Projection algorithm (AP) [6], RLS algorithm [4], and the Orthogonal Least Squares (OLS) algorithm [3]. BLS and AP are both underdetermined LS algorithms, both minimizing the LS problem over the last $F_a < M$ observations. Their main difference is that BLS makes an update every F_a steps (contigues window), whereas AP updates every step (sliding window). If we collect the F_a past input vectors in a $(F_a \times M)$ matrix $\underline{U}(k)$

$$\begin{aligned}\underline{U}(k) &= (\underline{u}(k), \underline{u}(k-1), ..., \underline{u}(k-F_a+1)) && (15)\\ &= \left(\underline{u}(k), \underline{\tilde{U}}(k)\right), && (16)\end{aligned}$$

the update equation in state-space form can be written as

$$\underline{\epsilon}(k{+}1) = \left(\underline{I} - \alpha \underline{U}(k) \left(\underline{U}^T(k)\underline{U}(k)\right)^{-1} \underline{U}^T(k)\right) \begin{cases} \underline{\epsilon}(k) & \text{AP} \\ \underline{\epsilon}(k-F_a) & \text{BLS.} \end{cases} \quad (17)$$

For describing a LS problem $\alpha = 1$, but in literature a smaller step-size $0 < \alpha \leq 1$ has also been used (Gauss-Newton type). Although the LS minimization property is lost, the algorithms behave still like a CM, which can be seen if $\underline{\epsilon}(k)$ is separated into one part that is a linear combination of $\underline{U}(k)$, say $\underline{U}(k)\underline{\beta}(k)$, and another part from the complementary space, say $\underline{r}(k)$. Thus $\underline{\epsilon}(k) = \underline{U}(k)\underline{\beta}(k) + \underline{r}(k)$ with $\underline{r}^T(k)\underline{U}(k) = \underline{0}$. Applying this to the state-space form (17) above, we get $\underline{\epsilon}(k{+}1) = (1-\alpha)\underline{U}(k)\underline{\beta}(k) + \underline{r}(k)$, whose norm is smaller then $\|\underline{\epsilon}(k)\|_2^2$ for $0 < \alpha < 2$. For $\alpha = 1$ the LS solution at time k shows the well-known orthogonality property of the a-posteriori errors $\underline{\tilde{U}}^T(k)\underline{\epsilon}(k) = \underline{0}$. Thus, for the AP algorithm the term from (17) becomes

$$\underline{U}(k) \left(\underline{U}^T(k)\underline{U}(k)\right)^{-1} \underline{U}^T(k)\underline{\epsilon}(k) = \frac{\underline{\psi}(k)}{\|\underline{\psi}(k)\|_2^2}\underline{u}^T(k)\underline{\epsilon}(k)$$

with a vector $\underline{\psi}(k)$ that is a linear combination of the past input vectors

$$\underline{\psi}(k) = \underline{u}(k) - \underline{\tilde{U}}(k)\underline{a}(k)$$

with the property

$$\begin{pmatrix} \underline{u}^T(k) \\ \underline{\tilde{U}}^T(k) \end{pmatrix} \underline{\psi}(k) = \begin{pmatrix} \underline{\psi}^T\underline{\psi}(k) \\ \underline{0} \end{pmatrix}. \quad (18)$$

Thus, (17) can be rewritten as a symmetric algorithm

$$\underline{\epsilon}(k+1) = \underline{\epsilon}(k) - \frac{\underline{\psi}(k)\underline{\psi}^T(k)}{\underline{\psi}(k)\underline{\psi}^T(k)}\underline{\epsilon}(k). \quad (19)$$

The RLS algorithm is not much different from the AP and BLS algorithm. The direction $\underline{\psi}(k)$ is also chosen to be orthogonal to the past input vectors. As long as a set of F_a input vectors is not spanning the whole space $\boldsymbol{R}^M$, the direction $\underline{\psi}(k)$ can be strict orthogonal and the behavior is the same than that of an AP(F_a) algorithm. After the directions span the whole space, the orthogonality can only be achieved in a LS sense. An advantage of RLS is the normalized choice $\underline{u}^T(k)\underline{\psi}(k) = 1$, such that $0 < \alpha < 2$ can be chosen directly, leading to CM. The most amazing result here is that both algorithm types, symmetric and unsymmetric, can behave the same, as long as the unknown system does not change too quickly.

Like in AP where the direction $\underline{\psi}(k)$ is orthogonal to the F_a past input vectors $\underline{\psi}^T(k)\underline{\tilde{U}}(k) = 0$, it is also possible to choose the update vector to be orthogonal to the previous directions

$$\begin{aligned}\begin{pmatrix} \underline{u}^T(k) \\ \underline{\tilde{\psi}}^T(k) \end{pmatrix} \underline{\psi}(k) &= \begin{pmatrix} \underline{\psi}^T\underline{\psi}(k) \\ \underline{0} \end{pmatrix} && (20)\\ \underline{\tilde{\psi}}(k) &= \left(\underline{\psi}(k-1), \underline{\psi}(k-F_a)\right). && (21)\end{aligned}$$

The desired direction can be investigated by a recursive filter

$$\underline{\psi}(k) = \underline{u}(k) - \sum_{i=1}^{F_c} c_i\, \underline{\psi}(k-i). \quad (22)$$

Applying $\underline{\psi}(k)$ to both sides of (22) we directly obtain $\underline{\psi}^T(k)\underline{\psi}(k) = \underline{\psi}^T(k)\underline{u}(k)$ and applying $\underline{\psi}(k-l)$ we get the coefficients

$$c_l(k) = -\frac{\underline{u}^T(k)\underline{\psi}(k-l)}{\underline{\psi}^T(k)\underline{\psi}(k-l)}. \quad (23)$$

To understand the algorithm's behavior it is necessary to consider the first update equations. We initialize the recursive filter with zeros: $\underline{\psi}(-1) = \underline{\psi}(-2) = \underline{\psi}(-F_c) = 0$ and obtain the following relation when using $\alpha = 1$

$$\underline{\psi}^T(k-l)\underline{\epsilon}(k) = 0 \quad \text{for } l = 1..F_a. \quad (24)$$

Applying the instantaneous direction $\underline{\psi}(k)$ to the update equation and using the result (24), we again find the relation

$$\underline{\psi}^T(k)\underline{\epsilon}(k) = \underline{u}^T(k)\underline{\epsilon}(k), \quad (25)$$

which leads to a symmetric algorithm. Like the AP algorithm we obtain CM for $0 < \alpha < 2$. Similarities and differences in the behavior of both will be explained in the next section.

The procedure of deriving the directions in the last method can also be described as a Gram-Schmidt orthogonalizing procedure. If the procedure is continued to $F_a = M$, we obtain the OLS algorithm [3], which has proven to be efficient for radial basis neural networks. Finally, Table 2 lists common unsymmetric algorithms.

$\underline{\psi}^T(k)\underline{\tilde{U}}(k) = 0$	BLS and AP(F_a)
$\min\{\|\underline{\psi}^T(k)\underline{\tilde{U}}(k)\|_2^2\}$	RLS-algorithm
$\underline{\psi}^T(k)\underline{\tilde{\psi}}(k) = 0$	Orthogonal direction alg.
$\min\{\|\underline{\psi}^T(k)\underline{\tilde{\psi}}(k)\|_2^2\}$	OLS-algorithm
$\underline{\psi}^T(k)\underline{\tilde{U}}(k)] = 0$	Mboup algorithm [5]

Table 2: Special cases for unsymmetric algorithms

III. Specific results

For computing the convergence speed we assume that the following set of M vectors $\underline{u}_1..\underline{u}_M$ build an orthogonal basis. Then, every vector $\underline{\epsilon}(k)$ can be written in terms of its projections

$$\underline{\epsilon}(k) = \sum_{i=1}^{M} \frac{\underline{\epsilon}^T(k)\underline{u}_i}{\underline{u}_i^T\underline{u}_i}\underline{u}_i \quad (26)$$

Thus, the squared L_2–norm of $\underline{\epsilon}(k)$ can be written as

$$\|\underline{\epsilon}(k)\|_2^2 = \sum_{i=1}^{M} \frac{\left(\underline{\epsilon}^T(k)\underline{u}_i\right)^2}{\underline{u}_i^T\underline{u}_i}. \quad (27)$$

If we assume that all of the M vectors are equally likely, we can assume that every term of the sum has an average value of $\|\underline{\epsilon}(k)\|_2^2/M$, or eqivalently, $\kappa_{\psi}^2(k) = \kappa_u^2(k) = 1/M$. Applying this to the LMS (NLMS) case, we obtain

$$C_{LMS}(k) = 1 - \frac{\mu(k)\|\underline{u}(k)\|_2^2(2-\mu(k)\|\underline{u}(k)\|_2^2)}{M} \quad (28)$$

$$C_{NLMS}(k) = 1 - \frac{\alpha(2-\alpha)}{M} \,. \quad (29)$$

In fact, equations (28) and (29) describe the convergence speed for a white input process very well. For white input processes we can assume that the instantaneous input vector is always the first of a set of orthogonal vectors, whereas for a correlated input sequence the vectors can no longer build an orthogonal set and the convergence speed becomes smaller. Since (N)LMS uses only $\underline{\epsilon}(k)$ and $\underline{u}(k)$, the tracking and initial behavior of this algorithm is the same.

This is different for the affine projection algorithm, where $\underline{\psi}(k)$ is chosen to be orthogonal to past F_a input vectors. During the initial phase the first direction is chosen to be $\underline{\psi}(0) = \underline{u}(0)$, whereas the next is orthogonal to it: $\underline{\psi}^T(1)\underline{u}(0) = 0$. Since the directions at time k are linear combinations of the instantaneous and the past $F_a - 1$ input vectors, $\underline{\psi}^T(i)\underline{\tilde{U}}(i) = 0$ includes also $\underline{\psi}^T(i)\underline{\psi}(i-l) = 0$ for $l = 1..F_a - 1$. The initial update sequence can be described as

$$\underline{\epsilon}(l) = \underline{\epsilon}(0) - \alpha \sum_{i=1}^{l} \frac{\underline{\psi}(i)\underline{\psi}^T(i)}{\underline{\psi}^T(i)\underline{\psi}(i)} \underline{\epsilon}(0) \,; l = 1..F_a \,. \quad (30)$$

Thus, the convergence speed, given by the quotient of the $i+1$–th and the i–th estimate is

$$C_{AP}(i) = 1 - \frac{\alpha(2-\alpha)}{M - i\alpha(2-\alpha)}; \text{ for } i = 1..F_a \quad (31)$$

However, the tracking behavior is completely different. We assume the step-size $\alpha = 1$ and obtain (19). If we compare this update equation with those of the NLMS algorithm, we can directly conclude that the AP algorithm of order F_a performs the same convergence speed as an NLMS algorithm for $\alpha = 1$ whose input sequence has been decorrelated by a filter of order F_a. The algorithm that uses orthogonal directions instead has a faster convergence speed. Substituting the past $P \leq F_a$ updates, we obtain

$$\underline{\epsilon}(k+1) = \underline{\epsilon}(k-P) - \alpha \sum_{i=0}^{P} \frac{\underline{\psi}(k-i)\underline{\psi}^T(k-i)}{\underline{\psi}^T(k-i)\underline{\psi}(k-i)} \underline{\epsilon}(k-P) \quad (32)$$

Applying the same argument as in the LMS-case, but taking (20) into account, we get

$$C_{OD}^{P} \simeq 1 - \frac{\alpha\,(2-\alpha)\,P}{M} \qquad ; \text{ for } P = 1..F_a \quad (33)$$

which describes $C(k)$ for P steps. If we calculate the quotient of $C_{OD}^{F_a}$ and $C_{OD}^{F_a - 1}$ we get the desired factor

$$C_{OD} \simeq \frac{\|\underline{\epsilon}(F_a)\|_2^2}{\|\underline{\epsilon}(F_a - 1)\|_2^2} = 1 - \frac{\alpha(2-\alpha)}{M - F_a\alpha(2-\alpha)} \quad (34)$$

which is valid for every MA random process with order lower or equal to F_a.

If F_a is set equal to the filter order M, the RLS algorithm with rectangular window is obtained from the AP algorithm. The convergence behavior of the algorithm is now essentially described by its initialization phase. Since $\alpha = 1$, the factor changes to $1/(M-i)$ with every time step and, therefore

$$C_{RLS}(i) \simeq 1 - \frac{1}{M-i} \qquad ;\text{for } i = 0..M-1. \quad (35)$$

After the initialization phase the weight-error vector is minimized and as long as the unknown system remains constant the algorithm only minimizes additional noise. If the system changes abruptly, the tracking behavior is poor, since the algorithm includes all former (wrong) observations.

The deviation of the CM property assumed an unknown system that is at least $F = \max\{F_a, F_b\}$ steps time-invariant. If a time-varying system can be described as approximately time-constant for F succeeding steps, than the fastest possible algorithm uses the F past observations, resulting in

$$C_{FAST} = 1 - \frac{1}{M-F} \,. \quad (36)$$

We finally like to note that if noise is included the approach still holds for CM in the mean.

References

1. N. J. Bershad, "Analysis of the normalized LMS algorithm with Gaussian inputs," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 34, no. 4, pp. 793–806, Aug. 1986.
2. E. Bjarnason, "Noise cancellation using a modified form of the filtered–XLMS algorithm," *Proc. Eusipco Signal Processing V,* Brussels, 1992.
3. S. Chen, S. McLaughlin, B. Mulgrew, "Complex-valued radial basis function network, part I: network architecture and learning algorithms," *Signal Processing,* vol. 35, no. 1, pp. 19-31, Jan. 1994.
4. S. Haykin, *Adaptive Filter Theory,* Englewood Cliffs, NJ: Prentice–Hall, Inform. and Syst. Sc. Series, 1986.
5. M. Mboup, M. Bonnet, "Une nouvelle structure blanchissante pour annulation d'echo acoustique," *Treizieme Colloque Gretsi,* Juan-Les-Pin du 16 au 20 Sep., 1991.
6. K. Ozeki, T. Umeda, "An adaptive filtering algorithm using orthogonal projection to an affine subspace and its properties," *Electronics and Communications in Japan,* vol. 67–A, no. 5, pp. 19–27, 1984.
7. M. Rupp, "The behavior of LMS and NLMS algorithms in the presence of spherically invariant processes," *IEEE Trans. Signal Processing,* vol. 41, no. 3, pp. 1149-1160, 1993.
8. M. Rupp, R. Frenzel "The behavior of LMS and NLMS algorithms with delayed coefficient update in the presence of spherically invariant processes," *IEEE Trans. Signal Processing,* vol. 42, no. 3, pp. 668-672, Mar. 1994.
9. B. Hassibi, A. H. Sayed, T. Kailath, "LMS is H^{∞} optimal," *IEEE CDC,* San Antonio, Texas, 1993.
10. U. Schultheiß, Über die Adaption eines Kompensators für akustische Echos, *Fortschrittberichte VDI,* 1988.