MULTI-STAGE ADAPTIVE SIGNAL PROCESSING ALGORITHMS

Suleyman S. Kozat and Andrew C. Singer

University of Illinois at Urbana-Champaign
Urbana, IL 61801 USA
Email: {*kozat, singer*}*@ifp.uiuc.edu*

ABSTRACT

In this paper, we explore the use of multi-stage adaptation algorithms for a variety of adaptive filtering applications where the structure of the underlying process to be estimated is unknown. These algorithms are "multi-stage" in that they comprise multiple adaptive filtering algorithms that operate in parallel on the observation sequence, and adaptively combine the outputs of this first stage to form an overall signal estimate. Several examples of this class of algorithms are demonstrated and analyzed in both a deterministic and stochastic context with respect to their convergence and mean squared error. The first example of this class, a "universal" linear predictor, was recently introduced and shown to asymptotically achieve the performance of the best linear predictor for each sequence, (up to some maximal order). Two new algorithms have been developed that generalize this universal linear predictor, and explore the use of the LMS algorithm in each stage of adaptation. Each of these algorithms are compared through theoretical analysis of their behavior.

1. INTRODUCTION

The idea of multi-stage adaptive filtering is related to a number of emerging techniques in robust adaptive control [8], machine learning [5] and data compression [7]. While the optimal filtering structure for a wide-sense stationary observation sequence contains a single filter, the number of filter parameters and the optimal values of these parameters are usually unknown a priori. In fact, for non-stationary observation sequences, the number and values of these optimal parameters will generally be time varying. Even when the number of the parameters is assumed known or estimated before an adaptation algorithm is used, the problem of selecting the appropriate adaptation algorithm remains significant, since each adaptation algorithm has different learning and convergence behavior. While the least mean square (LMS) algorithm has recently been shown to be optimum in a certain L_∞ sense [6], and can often track time variations better than a recursive least squares (RLS) algorithm, the RLS algorithm is optimum for a number of statistical and deterministic criteria [3]. Thus, fixing a specific filter (or adaptation algorithm) has potentially significant drawbacks due to the lack of a priori information about the observation sequence. A multi-stage algorithm attempts to overcome these problems by combining multiple candidate adaptation algorithms, with the goal of sequentially achieving the performance of the best algorithm among them.

In this paper, we investigate two-stage adaptation algorithms, and note that multi-stage algorithms can be extended in a similar way. As shown in Figure 1, the first stage of a multi-stage adaptation algorithm consists of m different adaptation algorithms that operate in parallel on the observation sequence. At any time, the jth algorithm outputs $\hat{d}_j(n)$, which is compared with the observed (desired) data $d(n)$, and the error, $e_j(n)$, is fed back to the adaptation algorithm. Each algorithm operates in parallel, with their adaptation processes and outputs decoupled from each other. Depending on the application, these algorithms may include a wide variety of models. For example, they can be: m different linear predictors of order from 1 to m; different algorithms for direction of arrival estimation; or different adaptation algorithms (LMS, LMF, RLS,...) for the same filter structure in attempt to exploit the different convergence characteristics of each algorithm.

The second stage of the multi-stage algorithm is the model mixture stage. In this stage, the outputs of the first stage algorithms are adaptively combined to give the final response. The first stage outputs can be combined in terms of their performance on the observed data so far, or another adaptation algorithm can be run on the outputs of the first stage algorithms. When the algorithms are adaptively combined according to a performance-weighted combination, the resulting approach is similar to a Bayesian mixture, and closely related to a variety of methods used in machine learning and universal data compression, [1]. In a recent paper [1], an approach using RLS linear predictors of varying order in the first stage, and a Bayesian-type mixture in the second stage is shown to be "universal" in a least-squares sense with respect to both model orders and model parameters.

The organization of this paper is as follows. In Section 2, after describing the universal linear predictor, we state some prior results for the deterministic and stochastic context. The structure of the universal linear predictor (ULP) is important, since it provides a framework for a large class of multi-stage algorithms. In the third section, we introduce a variant of the ULP, the LMS-Bayesian multi-stage adaptation algorithm. This algorithm differs from the ULP in the first stage, such that the LMS algorithm is used instead of the RLS algorithm to update the coefficients of the linear predictors. The convergence characteristics of the LMS-Bayesian algorithm are then analyzed for Gaussian data. In Section 4, the LMS-LMS multi-stage algorithm is defined and analyzed in this stochastic context. The LMS-LMS algorithm uses the LMS adaptation algorithm in both the first and second stages. Using independence assumptions, we explore the convergence behavior of the LMS-LMS algorithm.

2. UNIVERSAL LINEAR PREDICTOR

In this section, we describe the "universal" linear predictor, which has recently been introduced [1]. Let $\hat{x}_k(n)$ be the output of a sequential linear predictor as obtained by the RLS algorithm with model order k. Define a universal linear predictor as a weighted sum over linear predictors of order less than or equal to m,

$$\hat{x}_u(n) = \sum_{k=1}^{m} u_k(n)\hat{x}_k(n), \qquad (1)$$

0-7803-6339-6/00/$10.00 ©2000 IEEE

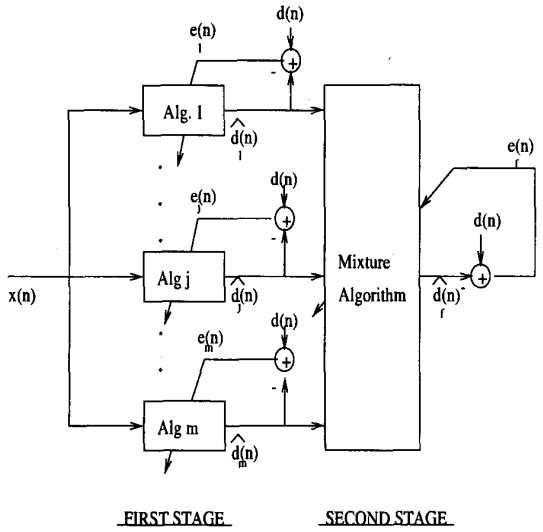

Figure 1: Multi-stage Adaptation Algorithm

$$u_k(n) = \frac{exp(-c\, l_{n-1}(x, \hat{x}_k))}{\sum_{j=1}^{m} exp(-c\, l_{n-1}(x, \hat{x}_j))},$$

where c is a positive constant and $u_k(n)$, the weights in the mixture, are proportional to the performance of the kth order predictor on the data observed so far. The performance, $l_{n-1}(x, \hat{x}_k)$ is the accumulated squared prediction error that results from sequential application of the time varying set of predictor coefficients, $w_{1,k}^n, \ldots, w_{k,k}^n$, i.e., by using $\hat{x}_k(n)$. For each new sample at time n, these coefficients are obtained such that the total squared prediction error,

$$E_{n-1}(x, \hat{x}_k) = \sum_{t=1}^{n-1} \left(x(t) - \sum_{i=1}^{k} w_{i,k} x(t-i) \right)^2, \quad (2)$$

is minimized over these coefficients. The accumulated squared error is then given by,

$$l_n(x, \hat{x}_k) = \sum_{t=1}^{n} \left(x(t) - \sum_{j=1}^{k} w_{j,k}^{t-1} x(t-j) \right)^2.$$

Because these linear prediction coefficients are optimized only over the data available (up to but not including the value to be predicted), the sequential prediction error is a "fair" measure of performance of each predictor.

The accumulated average square error of this algorithm is better, to within a negligible term, than that of an RLS predictor whose order is preset to the best order, say p, where $p \leq m$. Since the RLS algorithm of order p asymptotically achieves the performance of any fixed linear predictor of order p, this algorithm asymptotically attains the performance of the best fixed linear predictor of any order less than some m. The only assumption needed for this result is that the predicted sequence is bounded, i.e., $|x(n)| \leq A$, but is otherwise an arbitrary real-valued sequence. Then, the performance of this predictor can be related to that of the best sequential and batch predictors of order less than m, by Theorem-1 and Corollary-1 of [1] which is given by,

$$\frac{1}{n} l_n(x, \hat{x}_u(n)) \leq \min_k \frac{1}{n} l_n(x, \hat{x}_u(n)) + \frac{8A^2}{n} \ln(m),$$

and, corollary

$$\frac{1}{n} l_n(x, \hat{x}_u(n)) \leq \min_k \left\{ \frac{1}{n} E_n(x, \hat{x}_p(n)) + \frac{8A^2}{n} \ln(m) + O\left(\frac{\ln(n)}{n} \right) \right\}.$$

Thus the average squared prediction error of the universal prediction algorithm is within $O(n^{-1})$ of the best sequential linear prediction algorithm and within $O(n^{-1} \ln(n))$ of the best batch linear prediction algorithm, uniformly for every individual sequence. The cost terms can be identified as a model redundancy term proportional to $n^{-1} \ln(m)$ due to the lack of knowledge of the best model order, plus a parameter redundancy term proportional to $n^{-1} \ln(n)$ due to the lack of knowledge of the parameters and the learning time of RLS.

In a probabilistic setting, say for Gaussian data, it can be shown [4] that the universal linear predictor is convergent in the mean value, with a few plausible assumptions whose affects diminish with n. When the best order for prediction is p with $p \leq m$, we have

$$\lim_{n \to \infty} E[\mu_k(n)] = \begin{cases} 0 & k \neq p \\ 1 & k = p, \end{cases}$$

and when $p > m$,

$$\lim_{n \to \infty} E[\mu_k(n)] = \begin{cases} 0 & k \neq m \\ 1 & k = m. \end{cases}$$

It also can be shown that the learning curve of this universal linear predictor can be approximated as a weighted sum over the learning curves of all predictors used in the algorithm. As n goes to infinity, the MSE of the universal linear predictor converges to the MSE of the best order linear predictor used in the algorithm, i.e.,

$$E[(x(n) - \hat{x}_u(n))^2] \to \min_{k=1,\ldots,m} E[(x(n) - \hat{x}_k(n))^2], \quad (3)$$

Therefore, the universal linear predictor is universal in a stochastic sense as well, such that it achieves the MSE performance of the best predictor for the observed data up to some order m.

3. LMS-BAYESIAN

The linear predictors used in first stage of the universal linear predictor, defined in Section 2, minimize the total squared prediction error, over the previously observed data, i.e. eq. (2). In this section, the weights in the first stage of the algorithm will be updated by using the well-known least mean square (LMS) recursion. The derivations can be generalized to apply to the least mean fourth (LMF) or other gradient decent algorithms in a straightforward manner.

Let $\hat{x}_k(n)$ be the output of a sequential linear predictor as obtained by the least mean square (LMS) algorithm with model order k, i.e.,

$$\hat{x}_k(n) = \underline{w}_k^T(n) \underline{x}_k(n), \quad e_k(n) = x(n) - \hat{x}_k(n),$$

$$\underline{w}_k(n+1) = \underline{w}_k(n) + \mu e_k(n) \underline{x}_k(n), \quad (4)$$

where, μ is a constant to control the stability and the rate of convergence, $e_k(n)$ is the sequential error to be minimized in the mean

square, $\underline{w}_k(n) = [w_{1,k}^{n-1}, \ldots, w_{k,k}^{n-1}]^T$, and $\underline{x}_k(n) = [x(n-1), \ldots, x(n-k)]^T$. Define a new predictor as a weighted sum over the linear predictors of order less than or equal to m, as in eq. (1), where c is a positive constant and $u_k(n)$, the weights in the mixture, are proportional to the performance of the kth order predictor on the data observed so far, $l_{n-1}(x, \hat{x}_k)$.

3.1. Convergence of Weight Coefficients in the mean

Suppose the underlying process to be estimated is a zero mean stationary Gaussian random process with unknown covariance. In this probabilistic setting, the expected squared prediction error of an LMS linear predictor of order p for any n approximately satisfies [3],

$$J_p(n) = J_{min,p}\left(1 + \sum_{k=1}^{p} \frac{\lambda_k \mu}{1 - 2\mu\lambda_k}\right) + \sum_{k=1}^{p} \beta_k \alpha_k^n, \quad (5)$$

where λ_k is the kth eigenvalue of the correlation matrix $E[\underline{x}_k(n)\underline{x}_k^T(n)]$, and $J_{min,p}$ exists and is the optimal expected square error for the pth order linear predictor. The terms, α_k, are the eigenvalues of a related matrix [3]. The quantity $J_{min,p}$ is a non-increasing function of p such that the optimal pth order linear predictor asymptotically outperforms (or at least gives the same minimum error of) any predictor with order less than p. For small values of μ, eq. (5) can be approximated such that,

$$1 + \sum_{k=1}^{p} \frac{\lambda_k \mu}{1 - 2\mu\lambda_k} \approx 1 + \sum_{k=1}^{p} \lambda_k \mu,$$

which is equal to $(1+\mu p\sigma_x^2)$. When the first stage LMS algorithms are convergent, it can be verified that $|\alpha_k| \leq 1$ [3]. With this condition, the accumulated mean-squared prediction error of an LMS algorithm of order p will be approximately,

$$E[l_n(x, \hat{x}_p)] = \sum_{l=1}^{n} J_p(l) = J_{min,p} n(1 + \mu p\sigma_x^2)$$

$$+ \sum_{l=1}^{n}\sum_{k=1}^{p} \beta_k \alpha_k^l \leq J_{min,p} n(1 + \mu p\sigma_x^2) + \frac{p\beta^*\alpha^*}{1 - \alpha^*}, \quad (6)$$

where $\beta^* = \max_k \beta_k$ and $\alpha^* = \max_k \alpha_k$. Since the sum of the geometric terms are $o(n)$, their contributions will be negligible in comparison to the terms linear in n. For calculation of the mean values of the mixture coefficients, $u_p(n)$, we make the assumption that, asymptotically,

$$E[u_p(n)] \approx \frac{\exp(-c\, E[l_{n-1}(x, \hat{x}_p)])}{\sum_{k=1}^{m} \exp(-c\, E[l_{n-1}(x, \hat{x}_k)])}.$$

Then by eq. (6),

$$E[u_p(n)] \approx \frac{1}{1 + \sum_{k=1, k\neq p}^{m} \exp(-c\, A_k\, n)}, \quad (7)$$

where,

$$A_k = J_{min,k}(1 + \mu k\sigma_x^2) - J_{min,p}(1 + \mu p\sigma_x^2).$$

Suppose the underlying process to be estimated, $x(n)$, is a wth order Gaussian AR process,

$$x(n) = \sum_{k=1}^{w} c_k x(n-k) + \varepsilon(n), \quad (8)$$

where $\varepsilon(n)$ is a sequence of i.i.d. Gaussian random variables with zero mean and variance σ_ε^2. When $m > w$, the term $J_{min,p}$ is a monotonically non-increasing function of p. For sufficient order predictors ($p \geq w$), we have $J_{min,p} = J_{min,w} = \sigma_\varepsilon^2$. Then, for any predictor with order $p > w$, at least one of the exponentials in the denominator will diverge due to the positive sign of the term $\mu(p-w)\sigma_x^2$, yielding,

$$\lim_{n\to\infty} E[u_p(n)] = 0, p = w+1, \ldots, m.$$

For a $p = w$th order predictor, the contributions (to the denominator) of the higher order terms will vanish as n increases, due to the negative sign of the linear term $\mu(k-p)\sigma_x^2$. Nevertheless, the contributions of lower order terms are subtle. Although, $J_{min,k}$ strictly decreases in $k < p$, the linear term $\mu k\sigma_x^2$ increases in k. Thus, to guarantee convergence in the mean, certain conditions must be imposed on μ, to balance these counteracting terms. From eq. (7), we observe that convergence is achieved when $J_{min,p}(1 + \mu p\sigma_x^2) < J_{min,k}(1 + \mu k\sigma_x^2)$ for any $k < p$. Then, we must have

$$\frac{(1 + \mu k\sigma_x^2)}{(1 + \mu p\sigma_x^2)} > \frac{\sigma_\varepsilon^2}{\sigma_x^2 - \underline{p}_k{}^T R_k \underline{p}_k} \triangleq a, \quad (9)$$

where $\underline{p}_k = E[x(n)\underline{x}_k(n)]$ is the cross correlation vector, and $R_k = E[\underline{x}_k(n)\underline{x}_k^T(n)]$ is the correlation matrix of the input, which is assumed to be positive definite. Because $J_{min,k}$ is monotonically decreasing in k for $k \leq p$, [3], the ratio a on the right hand side of eq. (9) is always between zero and 1. Since, the left hand side of eq. (9) is also monotonically decreasing from 1 to k/p, as μ increases from 0 to ∞, there will always be a nontrivial interval such that the condition given in eq. (9) is satisfied. The interval is given by,

$$\frac{1 - a}{p\sigma_x^2 a - k\sigma_x^2} > \mu > 0,$$

which provides,

$$\lim_{n\to\infty} E[u_p(n)] = 1.$$

Since by definition, $\sum_{k=1}^{m} E[u_k(n)] = 1$, we conclude that,

$$\lim_{n\to\infty} E[u_k(n)] = 0,\ k = 1, \ldots, w-1.$$

3.2. Mean Squared Error

It can be verified that the MSE of the LMS-Bayesian algorithm approximately satisfies,

$$\begin{aligned} J_u(n) &= E[(x(n) - \sum_{k=1}^{m} u_k(n)\hat{x}_k(n))^2], \\ &= \sum_{k=1}^{m} E[u_k(n)^2] J_k(n) \quad (10) \\ &\quad + \sum_{k=1}^{m}\sum_{k\neq l=1}^{m} E[u_k(n)u_l(n)]E[e_k(n)e_l(n)]. \end{aligned}$$

Eq. (10) can be derived using similar assumptions as those taken in traditional analysis of the LMS algorithm [3]. Through analysis similar to that used for the ULP algorithm [4], and after some algebra, the MSE of the LMS-Bayesian algorithm can be shown to satisfy

$$E[(x(n) - \hat{x}_u(n))^2] \to \min_{k=1,\ldots,m} E[(x(n) - \hat{x}_k(n))^2]. \quad (11)$$

4. LMS-LMS

In this section we define the LMS-LMS algorithm. A two-stage mth order LMS adaptation algorithm is given by,

$$\hat{d}_f(n) = \underline{w}_u^T(n)\underline{x}_u(n), \;\; e_u(n) = d(n) - \hat{d}_f(n), \tag{12}$$

$$\underline{w}_u(n+1) = \underline{w}_u(n) + \mu_2 e_u(n)\underline{x}_u(n), \tag{13}$$

where $\underline{x}_u(n) = [\hat{x}_1(n), \hat{x}_2(n) \ldots, \hat{x}_m(n)]^T$, and for each k, $\hat{d}_k(n)$ is the output of the kth order LMS predictor given by,

$$\hat{d}_k(n) = \underline{w}_k^T\underline{x}_k(n), \;\; e_k(n) = d(n) - \hat{d}_k(n), \tag{14}$$

$$\underline{w}_k(n+1) = \underline{w}_k(n) + \mu_1 e_k(n)\underline{x}_k(n), \tag{15}$$

with $\underline{x}_k(n) = [x(n), \ldots, x(n-k+1)]$. Define a weight matrix $W(n)$ such that,

$$W(n) = \begin{bmatrix} \underline{w}_1^T(n) \\ \underline{w}_2^T(n) \\ \vdots \\ \underline{w}_m^T(n) \end{bmatrix}, \;\; \underline{x}_u(n) = W(n)\underline{x}_m(n),$$

where each $\underline{w}_k^T(n)$ has been padded with enough zeros to make their size $m \times 1$, and $W(n)$ is a lower triangular matrix. With this notation, the iterations for the multi-stage algorithm can be given in matrix form as,

$$\underline{w}_u(n+1) = \underline{w}_u(n) + \mu_2 \left(d(n) - \underline{w}_u{}^T(n)W(n)\underline{x}_m(n)\right) W(n)\underline{x}_m(n), \tag{16}$$

$$W(n+1) = (W(n) + \mu_1(d(n)\underline{1} - W(n)\underline{x}_m(n)) \odot L, \tag{17}$$

where $\underline{1} = [1 \ldots 1]^T$ with size $m \times 1$, and L is a lower triangular matrix of ones including the diagonal terms. The operator '$\odot$' implies term-by-term matrix multiplication operator, i.e. $(A \odot B)[i,j] = A_{i,j}B_{i,j}$. The application of the L matrix to the right hand side of (17) is required so that the recursion will generate the lower triangular matrix $W(n+1)$.

4.1. Convergence of $\underline{w}_u(n)$ in the mean value

In this subsection the convergence characteristics of the LMS-LMS algorithms are derived by using the following independence assumptions

A1) $\underline{w}_k(n)$ is independent of $\underline{x}_k(n)$ for any k.

A2) $\underline{w}_u(n)$ is independent of $\underline{x}_u(n)$.

Taking the expectation of eq. (16), and using these assumptions yields,

$$\begin{aligned} E[\underline{w}_u(n+1)] &= E\left[\left(I - \mu_2 W(n)\underline{x}_m(n)\underline{x}_m^T(n)W^T(n)\right)\underline{w}_u(n) \right. \\ &\left. +\mu_2 d(n)W(n)\underline{x}_m(n)\right], \\ &= (I - \mu_2 E[W(n)RW^T(n)])E[\underline{w}_u(n)] + \mu_2 W_0(n)\underline{p}_m, \end{aligned} \tag{18}$$

where $W_0(n) = E[W(n)]$, and it is assumed that the limit $\lim_{n\to\infty} W_0(n) = W_0$ exists and is given by the Wiener solution. The vector $\underline{p}_m = E[d(n)\underline{x}_m(n)]$ is the cross correlation vector for mth order linear predictor. If we define the matrix $K(n) \triangleq E[W(n)RW^T(n)]$, the weight vector $\underline{w}_{u0}(n) \triangleq K^{-1}(n)W_0(n)\underline{p}_m$, and the weight error vector $\underline{\epsilon}(n) \triangleq \underline{w}_u(n) - \underline{w}_{u0}(n)$ then, the iteration (18) can be given by,

$$E[\underline{\epsilon}(n+1)] = (I - \mu_2 K(n))E[\underline{\epsilon}(n)]. \tag{19}$$

Through analogy to single stage adaptation algorithms, $K(n)$ can be identified as the time dependent correlation matrix for the multi-stage algorithm and $\underline{\epsilon}(n)$ corresponds to the weight error vector for $\underline{w}_u(n)$. The time-dependent matrix, $K(n)$, is positive semi-definite for all n when R is positive definite. If we impose the condition such that for any finite iteration n, $W(n)$ is a nonsingular matrix (for example by adding a small diagonal load), then $K(n)$ is a positive definite matrix. A sufficient condition for this time recursion to be convergent in the mean value can be given as,

$$0 < \mu_2 < \frac{2}{\lambda_{max}(K(n))}. \tag{20}$$

Since at each iteration, $\| E[\underline{\epsilon}(n+1)] \|_2 \leq (1 - \mu_2 \times \lambda_{min}(K(n)) \times \| E[\underline{\epsilon}(n)] \|_2$, the adaptation process will cause $E[\underline{\epsilon}(n)] \to 0$. The (k,j)th element of $K(n)$ is given by,

$$K(n)_{[k,j]} = E[\underline{w}_k^T(n)R\underline{w}_j(n)] = tr(E[\underline{w}_j(n)\underline{w}_k^T(n)]R).$$

When the algorithm is near convergence, it is reasonable to assume that,

A3) The weight vectors in the first stage of the algorithm are independent from each other.

With this assumption,

$$K(n)_{[k,j]} = E[\underline{w}_k^T(n)]RE[\underline{w}_j(n)], \; k \neq j,$$

and for $(k = j)$,

$$\begin{aligned} K(n)_{[k,k]} &= tr\left(E[\underline{w}_k(n)\underline{w}_k^T(n)]R\right) \\ &= tr\left(E[\underline{\epsilon}(n)\underline{\epsilon}^T(n)]R\right) + 2\underline{w}_{k,0}^T RE[\underline{w}_k(n)] - \underline{w}_{k,0}^T R\underline{w}_{k,0}. \end{aligned} \tag{21}$$

The first quantity in eq. (21) is the time dependent excess mean squared error for kth order linear predictor [3]. As n increases, $K(n)$, the time dependent correlation matrix, converges to

$$\lim_{n\to\infty} K(n)_{[k,j]} = \begin{cases} \underline{w}_{j,0}^T R\underline{w}_{k,0} & k \neq j \\ \underline{w}_{k,0}^T R\underline{w}_{k,0} + J_{ex,k}(\infty) & k = j, \end{cases}$$

and $J_{ex,k}(\infty) = (J_{min,k})k\sigma_x^2$, is the excess mean squared error for kth order predictor where $J_{min,k}$ is the minimum mean squared error for kth order predictor. This equality can be written in compact form as

$$K \triangleq \lim_{n\to\infty} K(n) = W_0 R W_0^T + \Lambda_{ex},$$

where Λ_{ex} is a diagonal matrix, and $\Lambda_{ex,[k,k]} = J_{ex,k}(\infty)$. Thus, we conclude that $K(n)$ is convergent. By (19) and $E[\underline{\epsilon}(n)] \to 0$, we can find the vector that $E[\underline{w}_u(n)]$ converges as n increases, to

$$\begin{aligned} \lim_{n\to\infty} E[\underline{w}_u(n)] &= \lim_{n\to\infty} \underline{w}_{u0}(n), \\ &= \lim_{n\to\infty} K^{-1}(n)W_0(n)\underline{p}_m \\ &= K^{-1}W_0\underline{p}_m. \end{aligned} \tag{22}$$

Since the final output of the multi-stage algorithm is $\hat{d}_f(n) = \underline{w}_u^T(n)W(n)\underline{x}_m(n)$, the vector $\underline{w}_u^T(n)W(n)$ is the total coefficient vector that the multi-stage algorithm updates in two stages. Thus, it is instructive to observe $E[W^T(n)\underline{w}_u(n)]$. Since it is assumed that $\underline{w}_u(n)$ and $W(n)$ are independent,

$$E[W^T(n)\underline{w}_u(n)] = E[W^T(n)]E[\underline{w}_u(n)].$$

By using eq. (22) and K,

$$\lim_{n\to\infty} E[W^T(n)\underline{w}_u(n)] = W_0 K^{-1} W_0 \underline{p}_m.$$

If Λ_{ex} is negligible in comparison to $W_0 R W_0^T$ then,

$$\lim_{n\to\infty} E[W^T(n)\underline{w}_u(n)] = R^{-1}\underline{p}_m,$$

i.e. the overall algorithm converges to the Wiener solution.

4.2. Mean Squared Error

A time recursion for $W^T(n)\underline{w}_u(n)$ can be given by using eqs. (16) and (17). Expressing the MSE in closed form from these equations is complicated. Another approach for solving this problem is conditioning the MSE on $\underline{w}_u$ (i.e. not on W) so that,

$$E[(d(n) - \hat{d}_f(n))^2 | \underline{w}_u] = \sigma_d^2 - \underline{p}_m{}^T(n) K^{-1}(n) \underline{p}_m$$
$$+ (\underline{w}_u - \underline{w}_{u,0})^T K(n) (\underline{w}_u - \underline{w}_{u,0}), \qquad (23)$$

where $K(n) = E[W(n) R W^T(n)]$, $\underline{w}_{u,0} = K^{-1}(n)\underline{p}_m(n)$ and $\underline{p}_m(n) = E[d(n) W \underline{x}_m(n)] = W_0 \underline{p}_m$. For the last term in eq. (23),

$$E[(\underline{w}_u - \underline{w}_{u,0})^T K(n) (\underline{w}_u - \underline{w}_{u,0})]$$
$$= tr(E[(\underline{w}_u - \underline{w}_{u,0})(\underline{w}_u - \underline{w}_{u,0})^T] K(n)].$$

This is related to the well-known error covariance recursion for an ordinary mth order LMS algorithm. The only difference is the time dependence of the correlation matrix $K(n)$. We can define the weight error vector as $\underline{\epsilon}(n) = \underline{w}_u(n) - \underline{w}_{u,0}(n)$ and the error of the optimum Wiener filter as $e_{u,0}(n) = d(n) - \underline{x}_u^T(n)\underline{w}_{u,0}(n)$. Since the correlation matrix is time dependent, the optimal weight vector is also time dependent. By the orthogonality principle $e_{u,0}(n)$ is orthogonal to $\underline{x}_u(n) = W(n)\underline{x}_m(n)$, the input vector of the second stage. Thus after some algebra, the time recursion for the weight error covariance matrix $T(n) \triangleq E[\underline{\epsilon}(n)\underline{\epsilon}^T(n)]$ is given by,

$$T(n+1) = T(n) - \mu_2 T(n) K(n) - \mu_2 K(n) T(n)$$
$$+ \mu_2^2 [J_{min}(n) + tr(T(n) R(n))] K(n), \qquad (24)$$

where $tr(A) = trace(A)$, and
$J_{min}(n) \triangleq \sigma_d^2 - \underline{p}_m{}^T(n) K^{-1}(n) \underline{p}_m(n)$ is the time dependent MMSE for a given $K(n)$. Since $K(n)$ is Hermitian for all n, it can be decomposed as,

$$K(n) = Q(n)\Lambda(n)Q^T(n)$$

such that $Q^T(n)Q(n) = I$, and $\Lambda(n)$ is the diagonal matrix of time dependent eigenvalues of $K(n)$. If we make another assumption such that $Q(n)$ does not change from one iteration to another (or at least is slowly varying such that it is effectively constant), then we can diagonalize the recursion (24) by applying $Q(n)$ from both sides,

$$Q^T(n)T(n+1)Q(n) =$$
$$Q^T(n)T(n)Q(n) - \mu_2 Q^T(n)T(n)Q(n)\Lambda(n)$$
$$-\mu_2 \Lambda(n) Q^T(n) T(n) Q(n) + \mu_2^2 [J_{min}(n)$$
$$+ tr(Q^T(n)T(n)Q(n)\Lambda(n))]\Lambda(n).$$

If we define $R(n) = Q^T(n)T(n)Q(n)$, and $R(n+1) \approx Q^T(n)T(n)Q(n)$, then

$$R(n+1) = R(n) - \mu_2 R(n)\Lambda(n) - \mu_2 \Lambda(n) R(n)$$
$$+ \mu_2^2 [J_{min}(n) + tr(R(n)\Lambda(n))]\Lambda(n).$$

After some algebra, the conditions for this recursion to converge can be given in terms of the eigenvalues of $K(n)$ as, at each iteration,

$$0 \le \mu_2 \le \frac{1}{tr(K(n))}.$$

After convergence, the final mean squared error will be dependent upon $K = \lim_{n\to\infty} K(n) = W_0 R W_0^T + \Lambda_{ex}$, and $\lim_{n\to\infty} \underline{p}_m(n) = W_0 \underline{p}_m$. The final MSE is given by,

$$J(\infty) = J_{min}(\infty)(1 + \mu_2 tr(W_0 R W_0^T + \Lambda_{ex}),$$

where $J_{min}(\infty) = \sigma_d^2 - \underline{p}_m{}^T W_0^T (W_0 R W_0^T + \Lambda_{ex})^{-1} W_0 \underline{p}_m$. If Λ_{ex} is negligible in comparison $W_0 R W_0^T$, then $J_{min}(\infty) = \sigma_d^2 - \underline{p}_m{}^T R \underline{p}_m$, the MMSE for mth order optimum linear filter.

5. CONCLUSION

In this paper, we investigated a framework for analyzing multistage adaptation algorithms. Three examples of this class were analyzed in terms of their MSE and convergence characteristics. The MSE of the Universal and LMS-Bayesian algorithms are shown to converge to the MSE of the best predictor used. With some conditions on the adaptation constant, the final MSE of the LMS-LMS algorithm is calculated, and can asymptotically outperform any of the algorithms used in the first stage. Thus, all of these three algorithms are shown to be universal in this stochastic context.

6. REFERENCES

[1] A. C. Singer, M. Feder,"Universal Linear Prediction by Model Order Weighting," *IEEE Trans. on Signal Proc., vol. 47, no. 10, pp. 2685-2700, Oct. 1999.*

[2] M. Feder, A. C. Singer,"Universal Data Compression and Linear Prediction," *Proc. 1998 IEEE Data Compression Conference,1998.*

[3] S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, Upper Saddle River,NJ 07458,1996.

[4] S. S. Kozat, A. C. Singer,"On Universal Linear Prediction of Gaussian Data", *to appear Proc. Int. Conf. Acous. Speech and Sig. Process, Istanbul, 2000*

[5] D. Haussler, J. Kivinen, and M. Warmuth., "Sequential prediction of individual sequences under general loss functions," *IEEE Trans. Info. Theory*, vol. 44, pp. 1906–1925, Sept. 1988.

[6] B. Hassibi, A.H. Sayed, and T. Kailath, "H^∞ Optimality of the LMS Algorithm," *IEEE Transactions on Signal Processing*, vol. 44, no. 2, pp. 267–280, Feb. 1996.

[7] M. Weinberger, N. Merhav, and M. Feder, "Optimal sequential probability assignment for individual sequences,"*Trans. Info. Theory*, vol. 40, no. 2, pp. 384–396, March 1994.

[8] S. Kulkarni and P. Ramadge, "On the performance and complexity of a class of hybrid controller switching policies," in *Proc. Conf. on Control Using Logic-Based Switching*, (Block Island, RI), Springer-Verlag, 1995.