We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,300

Open access books available

172,000

International authors and editors

190M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Klaman Assisted LMS Methods

Nastoo Avesta
University of Ottawa
Canada

1. Introduction

The Least Mean Square (LMS) algorithm is a widely used adaptive method for identification, predication and noise suppression of Finite Impulse Response (FIR) filters (Haykin, 2002). Furthermore, it is extensively employed in various components of multimedia systems. Multimedia specific applications of LMS include, but certainly are not limited to, traffic routing (Xinyu et al, 2004), decision feedback channel equalization (Hoshino et al, 2000), signal classification (Bauckhage, 2005), echo cancelation (Sristi et al, 2001) and shape reconstruction (Mal-Rey, 2001). Main reasons for the popularity of LMS are its conceptual simplicity and ease of implementation. Although, these features are advantageous in both theoretical and practical developments; however, they tend to obscure the fine details of the systems under consideration. To address the shortcomings of a typical LMS algorithm, combined Klaman filter and LMS methods are proposed.

Klaman filter provides the optimal, in bayesian sense, state estimate of a linear system, based on degraded and noisy observations. It does so by taking into account the prescribed system model and simultaneously utilizing the difference between the predicated state and the observed one. The former, i.e., prescribed system model, is what LMS is incapable of taking into account, i.e., there are no means by which LMS can take into account partial observations or incomplete state measurements. Thus, under such circumstance, which are all too prevalent under any practical situation, LMS performance degrades rapidly. Although, recently, there have been other attempts to merge Kalman filtering and LMS algorithm (Lopez et al, 2007) and (Kohli & Mehra, 2007); however, these solutions relay on explicit assumptions on the observed system model. Methods described in this chapter are fundamentally different, as none make any implicit or explicit, assumptions on the observed system model. Here, we present full and partial Kalman filter based LMS update algorithms, where the traditional LMS algorithm is modeled as a noisy observation plant of a Kalman filter, with Kalman filter states representing the update coefficients. Nothing that multimedia applications often necessitate the use of long adaptive filters, requiring excessive computations to calculate the filter output and to update all coefficients, many researchers have addressed the impact of reducing the necessary computations through using shorter filters or updating only a subset of coefficients at every iteration. These partial update algorithms include periodic LMS update (Douglas, 1997), where every L-th coefficient is updated and the M-MAX algorithm (Mayyas & Aboulnasr, 2004), where the M coefficients with the most potential to reduce error are updated in a given iteration.

However, these partial update algorithms typically use the same update term that would have been used by the full update algorithm. Furthermore, no post processing or additional real-time processing is included to improve the overall estimation. Given the limited nature of partial updates, in this paper we propose an improved estimation of the full length filter, based on partial updates and observations.

The outline of this chapter is as follows. In Section 2, a brief description of gradient based adaptive methods, highlighting a number of performance criteria and stability requirements, is given. Section 3, provides an overview of Kalman filtering and its respective performance criterion. Klamman assisted methods for full and partial weight update are developed in Section 4 and Section 5. Finally, Section 6 provides a summary and an overview of the current and future line of research.

2. Gradient Based Adaptation

In this section, we consider two gradient based methods for adaptive filtering: Steepest Descent (SD) and Least Mean Square (LMS). We note that although SD method does not lend itself to practical implementation; however, it is capable of reaching the theoretical optimum solution. On the other hand, we also note that although LMS, readily, lends itself to practical implementation; however, its final solution may be far from the optimal solution. Developments of this section closely follow (Haykin, 2002).

2.1 Steepest Descent Method

At time instance n , let the output of an N tap adaptive filter, in response to input vector $\mathbf{x}(n)$, be $y(n)$, (1). In (1), the i^{th} element of $\mathbf{x}(n)$, $N \times 1$, is the tap delayed input $x(n-i)$ and $\mathbf{w}(n)$, $1 \times N$, represents the gain of each tap at instance n .

$$y(n) = \mathbf{w}(n)\mathbf{x}(n) \quad (1)$$

Let the desired response, at time n , be indicated by $d(n)$ and its difference with $y(n)$ by $e(n)$.

$$e(n) = d(n) - y(n) \quad (2)$$

Our aim is to minimize the Mean Square Error (MSE), (3), by judicial adjustment of the weight vector $\mathbf{w}(n)$, where, in (3), E is the expectation operator.

$$J(n) = E[e^2(n)] \quad (3)$$

Given that $J(n)$ is a quadratic function of $\mathbf{w}(n)$, i.e., $e(n)$ is a linear function of $\mathbf{w}(n)$ and elements of $\mathbf{w}(n)$ are assumed independent, a simple update policy is to move in the opposite direction of gradient vector of (3). Expanding (3) and differentiating it with respect to $\mathbf{w}(n)$, (4), the gradient vector is obtained, where $\mathbf{r}_{xd}$ is the cross correlation function of $\mathbf{x}(n)$ and $d(n)$; and $\mathbf{R}_{xx}$ the auto-correlation matrix of $\mathbf{x}(n)$.

$$\nabla J(n) = -2\mathbf{r}_{xd} + 2\mathbf{R}_{xx}\mathbf{w}(n) \quad (4)$$

The update policy may now be stated as (5), that is, the optimal solution is obtained by moving in the opposite direction of gradient vector, in step sizes equal to μ .

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu[\mathbf{r}_{xd} - \mathbf{R}_{xx}\mathbf{w}(n)] \quad (5)$$

It can be shown, e.g. see (Haykin, 2002), that for $\mathbf{w}(n)$ to converge to optimum solution as $n \rightarrow \infty$, μ must satisfy (6), where λ_{max} is the largest eigenvalue of $\mathbf{R}_{xx}$.

$$0 < \mu < \frac{2}{\lambda_{max}} \quad (6)$$

It is noted that for the SD method to reach to the optimum solution, both $\mathbf{R}_{xx}$ and $\mathbf{r}_{xd}$ must be known a priori.

2.2 Least Mean Square Algorithm

In this section, we provide an overview of the LMS algorithm, along with its performance measure characteristics. Let the noise induced observation, at time n , be indicated by $d(n)$, where the observation noise is characterized as a zero mean white Gaussian noise with variance σ_o^2 , $o \sim N(0, \sigma_o^2)$.

Adjusting $\mathbf{w}$ to minimize the instantaneous squared error, $e^2(n)$, we proceed by minimizing $e^2(n)$ as a function of $\mathbf{w}$.

$$\begin{aligned} \frac{de^2(n)}{dw_i} &= 0 \quad i = 0 \cdots N-1 \\ -2e \frac{dy(n)}{dw_i} &= 0 \end{aligned} \quad (7)$$

Noting that $y(n) = \sum_{k=0}^{N-1} w_k x(n-k)$ and hence $\frac{dy(n)}{dw_i} = x(n-i)$, the update equation (8) is obtained. That is, $\mathbf{w}$ is adjusted by moving in the opposite direction of instantaneous error surface gradient, in step sizes equal to μ .

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\mathbf{x}(n) \quad (8)$$

For LMS algorithm to converge to its final solution, μ must satisfy the bounds of (9), where σ_x^2 is the variance of the input signal.

$$0 < \mu < \frac{2}{N\sigma_x^2} \quad (9)$$

The performance of LMS is greatly influenced by its design parameter μ , for example, the minimum achievable MSE is directly proportional to μ .

$$J(\infty) = \sigma_o^2 \left(1 + \frac{\mu}{2} N \sigma_x^2 \right) \quad (10)$$

The performance criteria of LMS is measured based on Mean Square Error (MSE), as defined in (3), and Mean Square Deviation (MSD), as defined in (11), where $\varepsilon = \mathbf{w}^{Optimum} - \mathbf{w}$, $\mathbf{w}^{Optimum}$ is the optimum weight vector and $\|\cdot\|^2$ is the Euclidean norm.

$$D(n) = E \left[\|\varepsilon(n)\|^2 \right] \quad (11)$$

Furthermore, to characterize the transient behavior and convergence of the LMS, we identify the following parameters: mean square excess, misadjustment, and average time constant. Both mean square excess, (12), and misadjustment, (13), are measures of LMS optimality.

$$J_{ex}(n) = J(n) - \sigma_o^2 \quad (12)$$

$$M = J_{ex}(\infty) / \sigma_o^2 \quad (13)$$

Average time constant is a measure of convergence time of LMS. Average time constant can be estimated as in, (14), where λ_{av} is the average value of eigenvalues of $\mathbf{R}_{xx}$.

$$\tau_{mse, av} = \frac{1}{2\mu\lambda_{av}} \quad (14)$$

Noting that misadjustment may also be expressed as, (15), it becomes obvious that in dealing with traditional LMS a choice has to be made with respect to convergence time verses the final error, i.e., for faster convergence, larger μ , a larger error will be incurred and vice versa.

$$M = \frac{\mu N \lambda_{av}}{2} \quad (15)$$

Under system identification setup, Fig. 1, our main interest is to minimize MSD, irrespective of the observation noise.

Fig. 1. System Identification Setup

3. Kalman Filter

Let the state evolution of a system be described by (16), where $\mathbf{x}_k$, $\mathbf{u}_k$, $\mathbf{w}_k$, all $N \times 1$, are state of the system, the deterministic control and the observation noise; $\mathbf{A}_k$, $\mathbf{B}_k$, and $\mathbf{C}_k$, all $N \times N$, are drift, diffusion and colouring matrices, respectively, at instance k .

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_k + \mathbf{C}_k \mathbf{w}_k \quad (16)$$

Given the observation plant (17), where $\mathbf{z}_k$, and $\mathbf{v}_k$, both $M \times 1$, are the observation vector and observation noise; $\mathbf{D}_k$, $M \times N$, degradation matrix, respectively, then it could be shown, (Lewis, 1986), that an optimal linear estimate of $\mathbf{x}_k$ can be obtained through Kalman filtering.

$$\mathbf{z}_k = \mathbf{D}_k \mathbf{x}_k + \mathbf{v}_k \quad (17)$$

In particular, for $\mathbf{x}_0 \sim N(E[\mathbf{x}_0], \mathbf{P}_{x_0})$, $\mathbf{w}_k \sim N(0, \mathbf{Q}_k)$, and $\mathbf{v}_k \sim N(0, \mathbf{R}_k)$, i.e., all Gaussian statistics, this estimate is unconditionally optimal. The Kalman algorithm is described in (18), where $\mathbf{P}_k$, $N \times N$, is the posteriori error covariance, $\mathbf{P}_k^-$, $N \times N$, is the a priori error covariance, $\mathbf{I}$, $N \times N$, is the identity matrix, $\mathbf{K}_k$, $N \times M$, is the Kalman gain matrix, $\hat{\mathbf{x}}_k$, $N \times 1$, is the posteriori state estimate and $\hat{\mathbf{x}}_k^-$ is the a priori state estimate, at time instance k .

Initialization

$$\mathbf{P}_0 = \mathbf{P}_{x_0} \quad \hat{\mathbf{x}}_0 = E[\mathbf{x}]$$

Time Update

$$\mathbf{P}_{k+1}^- = \mathbf{A}_k \mathbf{P}_k \mathbf{A}_k^T + \mathbf{C}_k \mathbf{Q}_k \mathbf{C}_k^T$$

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{A}_k \hat{\mathbf{x}}_k + \mathbf{B}_k \mathbf{u}_k \quad (18)$$

Measurement Update

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^- \mathbf{D}_{k+1}^T (\mathbf{D}_{k+1} \mathbf{P}_{k+1}^- \mathbf{D}_{k+1}^T + \mathbf{R}_{k+1})^{-1}$$

$$\mathbf{P}_{k+1} = (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{D}_{k+1}) \mathbf{P}_{k+1}^-$$

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_{k+1}^- + \mathbf{K}_{k+1} (\mathbf{z}_{k+1} - \mathbf{D}_{k+1} \hat{\mathbf{x}}_{k+1}^-)$$

In the context of system identification, the main goal is the ability to identify the state of the system at any given time. In terms of Kalman filter, this means that the system described by (16) and (17) should be observable, i.e., the observability matrix (19) should be of rank N .

$$\mathbf{O} = [\mathbf{D}_k \quad \mathbf{D}_k \mathbf{A}_k \quad \dots \quad \mathbf{D}_k \mathbf{A}_k^{N-1}]^T \quad (19)$$

The algorithm of (18) is appropriate for on-line applications, where the optimal estimate at time instance k is strictly based on observation up to k . However, for off-line applications it is advantageous to employ all observation points to provide an estimate for all time instances. This formulation is referred to as smoothing Kalman filter, and is described in

(20), where P_k^s , $N \times N$, x_k^s , $N \times 1$, and F_k , $N \times N$, are smoothing error covariance, estimated vector and Kalman gain matrices, at instance k . It is noted that before smoothing can be employed, Kalman filter is applied for the duration L and optimal state estimates along with their respective posteriori covariance matrices are recorded.

Initialization

$$\begin{aligned}
 P_L^s &= P_L \quad x_L^s = \hat{x}_L \\
 \text{Update } k &\in \{L-1 \dots 0\} \\
 x_k^s &= \hat{x}_k + F_k(x_{k+1}^s - \hat{x}_{k+1}^-) \\
 F_k &= P_k A_k (P_{k+1}^-)^{-1} \\
 P_k^s &= P_k - F_k(P_{k+1}^- - P_{k+1}^s)F_k^T
 \end{aligned} \tag{20}$$

4. Full Update Kalman Least Mean Square Algorithms for System Tracking

In this section, we present two Kalman filter based Least Mean Square (LMS) update algorithms, for both on-line and posthumous system tracking applications. In both cases, the traditional LMS algorithm is modeled as a noisy observation plant of a Kalman filter, with Kalman filter states representing the update coefficients. It is shown that the proposed update algorithms achieve superior performance compared to the traditional LMS and RLS, under diverse conditions, without compromising the convergence time. In particular, it is shown that the proposed on-line algorithm is capable of achieving lower Mean Square Deviation (MSD) bound than LMS. Fig. 2 describes the proposed combined algorithms setup.

Fig. 2. Combined Kalman LMS Setup

4.1 Time-invariant Identification

Consider a regular system identification setup using an FIR filter. Let the coefficients of the adaptive LMS FIR filter at instant k be given by w_k^{LMS} and let the coefficients of the unknown time invariant system be given by w_k^{Opt} . We can then express LMS coefficients as observation vector of a Kalman filter, whose state equation describes the unknown system

coefficients,(21), where (21.a) is the state evolution plant, similar to (16), (21.b) is the observation plant, similar to, (17), and ε_k^w is a zero mean Gaussian noise with variance σ_w^2 .

$$\begin{aligned} \mathbf{w}_{k+1}^{Opt} &= \mathbf{w}_k^{Opt} & \text{a)} \\ \mathbf{w}_k^{LMS} &= \mathbf{w}_k^{Opt} + \varepsilon_k^w & \text{b)} \end{aligned} \quad (21)$$

σ_w^2 may be calculated as follows. Firstly, it is noted that MSD of LMS is bounded by (22), (Haykin, 2002), where λ_{max} and λ_{min} are the maximum and minimum eigenvalues of the input signal correlation matrix.

$$\frac{J_{ex}(k)}{\lambda_{max}} \leq MSD_k \leq \frac{J_{ex}(k)}{\lambda_{min}} \quad (22)$$

Secondly, (13) may be rewritten as (23).

$$J_{ex}(\infty) = M\sigma_o^2 \quad (23)$$

Thirdly, for unit variance white noise, (24) holds.

$$\lambda_{min} = \lambda_{max} = \sigma_x^2 = 1 \quad (24)$$

Combining (22) through (24), expression (25) for σ_w^2 is obtained.

$$\sigma_w^2 = \frac{\mu N \sigma_o^2}{2} \quad (25)$$

It can be shown, (Lewis, 1986), that in time-invariant case, Kalman filtering amounts to a simple weighted average of the observed LMS weight vector. That is, assuming a typical Kalman update, the independence of the weights and an initial Gaussian distributed weight vector, $\mathbf{w}_0$, of mean $\bar{\mathbf{w}}$ and variance $\sigma_{w_0}^2$, after k iterations the following updates are obtained.

$$\begin{aligned} P_k^{i-} &= \frac{\sigma_{w_0}^2}{1 + (k-1) \left(\frac{\sigma_{w_0}^2}{\sigma_w^2} \right)} \\ P_k^i &= \frac{\sigma_{w_0}^2}{1 + k \left(\frac{\sigma_{w_0}^2}{\sigma_w^2} \right)}; \quad K_k^i = \frac{\frac{1}{\sigma_w^2}}{\frac{1}{\sigma_{w_0}^2} + \frac{k}{\sigma_w^2}} \\ \hat{w}_k^i &= \frac{1}{1 + k \left(\frac{\sigma_{w_0}^2}{\sigma_w^2} \right)} \left[\hat{w}_0^i + \left(\frac{\sigma_{w_0}^2}{\sigma_w^2} \right) \sum_{j=1}^k w_k^{iLMS} \right] \\ \forall i &\in \{1 \dots N\} \end{aligned} \quad (26)$$

where $P_k^{i-}, P_k^i, K_k^i, \hat{w}_k^i, \sigma_{w_0}^2$ and $\hat{w}_0^i$ are the i^{th} element of the a priori error covariance, posteriori error covariance, Kalman gain, optimal estimate, initial error covariance, and initial weight estimate, respectively. For a typical system tracking and identification, we

may assume zero prior knowledge, *i.e.* $\sigma_{w_0}^2 = \infty$ and $\hat{w}_0^i = 0$, in which case the Kalman optimal weight estimate simplifies to (27).

$$\begin{aligned} \hat{w}_k^i &= \hat{w}_{k-1}^i + \frac{1}{1+k} \left[w_k^{iLMS} - \hat{w}_{k-1}^i \right] \\ \forall i &\in \{1 \cdots N\} \end{aligned} \quad (27)$$

4.2 Time Variant System Tracking

Based on (27), it is clear that measurement contributions are ignored as k increases. Hence, system model of (21) would not be able to accommodate tracking of time-varying systems. To address this shortcoming, we choose the update scheme of (28) based on Steepest Descent (SD), (5), to replace (21.a). Under the system identification assumptions input signal correlation matrix, input vector and observation scalar are available. However, the input-observation cross correlation function, r_{xd}^k , must be estimated. For ease of implementation, let this estimate be a simple recursive averaging, as in (29), with α between 0 and 1.

$$\begin{aligned} w_k^{SD} &= w_{k-1}^{SD} + \mu \left[r_{xd}^k - R_{xx}^k w_{k-1}^{SD} \right] \\ &= \left(I - \mu R_{xx}^k \right) w_{k-1}^{SD} + \mu r_{xd}^k \end{aligned} \quad (28)$$

$$\hat{r}_{xd}^k = \alpha \hat{r}_{xd}^{k-1} - (1 - \alpha) x(k) d(k) \quad (29)$$

Similarly, let the variance of r_{xd}^k , $\sigma_{x^k d^k}^2$, be approximated by (30), with β between 0 and 1.

$$\sigma_{x^k d^k}^2 = \beta \sigma_{x^{k-1} d^{k-1}}^2 + (1 - \beta) [x(k) d(k)]^2 \quad (30)$$

It is noted that (Taylor, 1997)

$$\hat{r}_{xd}^k = r_{xd}^k + \varepsilon_k^{xd} \quad (31)$$

where ε_k^{xd} is approximated as a zero mean Gaussian noise with variance $\sigma_{\varepsilon_k^{xd}}^2$. $\sigma_{\varepsilon_k^{xd}}^2$ may be approximated by the first differential of (29) as in (32):

$$\begin{aligned} \sigma_{\varepsilon_k^{xd}}^2 &= \left(\frac{\partial \hat{r}_{xd}^k}{\partial x^k d^k} \sigma_{x^k d^k} \right)^2 \\ &= (1 - \alpha)^2 \left(\beta \sigma_{x^{k-1} d^{k-1}}^2 + (1 - \beta) [x(k) d(k)]^2 \right) \end{aligned} \quad (32)$$

Based on the above, the following state-space description is obtained.

$$\begin{aligned} w_k^{SD} &= \left(I - \mu R_{xx}^k \right) w_{k-1}^{SD} + \mu r_{xd}^k \\ w_k^O &= w_k^{SD} + \mu \varepsilon_k^{xy} \end{aligned} \quad (33)$$

where w_k^O is the observed weight vector. However, it is noted that, unlike (21), the Kalman setup of (33) does not take into account the observation noise. To remedy this, we combine the setup of (21) and (33) as in (34), where, by comparing (5) and (8), it is noted that w_k^{LMS} , the observation vector of this Kalman LMS-SD model, also requires less computational effort than that of a typical SD algorithm.

$$\begin{aligned} \mathbf{w}_k^{SD} &= \left(I - \mu \mathbf{R}_{xx}^k \right) \mathbf{w}_{k-1}^{SD} + \mu \mathbf{r}_{xd}^k + \mu \overline{\varepsilon_k^{xd}} \\ \mathbf{w}_k^{LMS} &= \mathbf{w}_k^{SD} + \varepsilon_k^w \end{aligned}$$

(34)

Furthermore, it is noted that since for a stable LMS both the drift matrix, $\mathbf{A}_k = \left(I - \mu \mathbf{R}_{xx}^k \right)$, and degradation matrix, $\mathbf{D}_k = I$, are full rank, the proposed Kalman filter is indeed observable. The setup of (34) can be employed either on-line, employing typical Kalman filter, or posthumously, using smoothing Kalman filter.

4.3 Verification and Simulation Results

The performance of the system described by (34) has been studied for unity input variance, N=50 randomly generated taps, $\beta=\alpha$, simulation length of 2000 steps, various observation noises and step sizes. For time varying systems, the optimum coefficients are assumed to be a constant, modulated by a sine wave, with a randomly generated frequency of variances 0.5 and 5, corresponding to slowly and rapidly varying systems, respectively. To gain further understanding of the performance of on-line Kalman LMS-SD method (KLMSSD) and posthumous Kalman LMS-SD method (PKLMSSD), comparison against both traditional LMS and RLS method are included. The results are summarized in Table . In the following simulations α , β and λ_{RLS} , RLS forgetting factor, are selected, in order to minimize the MSE of their respective adaptive algorithm. Fig. 3 shows a typical performance comparison of LMS, KLMSSD, PKLMSSD and RLS algorithms. It is noted that while Kalman based algorithms retain the tracking capability of LMS, unlike RLS, they do so without introducing excessive noise in their output. Here, the vertical lines indicate the instances that LMS, KLMSSD, PKLMSSD and RLS have detected the zeros crossing of the 2 representative system coefficients. The significant delayed response of the RLS is noted. The poor performance of RLS can be readily identified in its state-space form, (Haykin, 2002), (35).

$$\begin{aligned} \mathbf{w}_{k+1}^{RLS} &= \lambda_{RLS}^{-\frac{1}{2}} \mathbf{w}_k^{RLS} \quad 0 \leq \lambda_{RLS} \leq 1 \\ y_k &= \mathbf{x}_k^T \mathbf{w}_k^{RLS} + \lambda_{RLS}^{-\frac{k}{2}} \eta_k \end{aligned}$$

(35)

where T is the transpose operator, x is the Nx1 input vector and η is the zero-mean Gaussian observation noise, with variance σ_o^2 .

Algorithm	$\sigma_o=1$ High SNR						$\sigma_o=10$ Low SNR					
	Time Invariant		Slowly Varying		Rapidly Varying		Time Invariant		Slowly Varying		Rapidly Varying	
	MSE	MSD	MSE	MSD	MSE	MSD	MSE	MSD	MSE	MSD	MSE	MSD
LMS	1.82	122	19.8	144	1859	142	165	118	490	136	2351	138
	$\mu=0.012$		$\mu=0.028$		$\mu=0.028$		$\mu=0.014$		$\mu=0.028$		$\mu=0.028$	
RLS	1.09	141	11.7	141	609	137	109	148	248	144	1235	139
	$\lambda=1$		$\lambda=0.95$		$\lambda=0.9$		$\lambda=1$		$\lambda=0.95$		$\lambda=0.9$	
KLMSSD	1.30	128	4.36	133	176	136	73	118	145	126	1036	123
	$\alpha=0.995$		$\alpha=0.995$		$\alpha=0.99$		$\alpha=0.95$		$\alpha=0.995$		$\alpha=0.99$	
PKLMSSD	0.84	116	1.65	129	122	134	67	111	61	107	307	106

Table I LMS, RLS, KLMSSD and PKLMSSD Performance Table

Comparing the system plant processes of (35) and (34) it is evident that RLS has no direct means by which it would be able to track changes in the unknown system. Its state-space form essentially assumes that the current state is that of the optimal state plus observation noise; whereas, KLMSSD is able to detect and track changes in the unknown system due to changes in r_{xd}^k .

Fig. 3. LMS, Kalman LMS, Smoothing Kalman LMS and RLS Comparison for Slowly Varying System Identification

4.4 KLMSSD MSD BOUND

In this section, we provide an existence proof to show that MSD bound lower than LMS is achievable via KLMSSD. To investigate the MSD obtained by the KLMSSD, the covariance error matrix of (34) is analyzed for the wide sense stationary case. It is noted that as $k \rightarrow \infty$ then $P_k \rightarrow P_k^+ \rightarrow P$, where P_k is the a priori covariance error and P_k^+ is the posteriori covariance error of the Kalman filter. For a stabilized system, P may be expressed as (36), where Q , the expected value of (32), is given in (37).

$$P = \left\{ \left(I - \mu R_{xx} \right) \left[P - P \left(P - \frac{\mu N \sigma_o^2}{2} \right)^{-1} P \right] \left(I - \mu R_{xx} \right)^T \right\} + \mu Q \mu \quad (36)$$

$$\begin{aligned}
 Q &= E \left[(1-\alpha)^2 \left(\beta \sigma_{xd}^2 + (1-\beta) (\mathbf{x}^k d^k)^2 \right) \right] \\
 &= (1-\alpha)^2 \left(\beta \sigma_{xd}^2 + (1-\beta) E \left[(\mathbf{x}^k d^k)^2 \right] \right) \\
 &= (1-\alpha)^2 \left(\beta \sigma_{xd}^2 + (1-\beta) \sigma_{xd}^2 \right) \\
 &= (1-\alpha)^2 \sigma_{xd}^2
 \end{aligned} \tag{37}$$

Noting that $\mathbf{P}$ is a diagonal matrix with diagonal element p_i , (36) simplifies to (38), which can further be simplified to (39).

$$\begin{aligned}
 p_i &= (1-\mu\sigma_x^2)^2 \left[p_i - \frac{p_i^2}{\left(p_i - \frac{\mu N \sigma_o^2}{2} \right)} \right] + \mu^2 (1-\alpha^2) \sigma_{xd}^2 \\
 &= \frac{\left[(1-\mu\sigma_x^2)^2 p_i \left(p_i - \frac{\mu N \sigma_o^2}{2} \right) - (1-\mu\sigma_x^2)^2 p_i^2 + \left(p_i - \frac{\mu N \sigma_o^2}{2} \right) \mu^2 (1-\alpha^2) \sigma_{xd}^2 \right]}{p_i - \frac{\mu N \sigma_o^2}{2}}
 \end{aligned} \tag{38}$$

$$\forall i \in \{1 \dots N\}$$

$$\begin{aligned}
 p_i^2 - p_i b_i + c_i &= 0 \\
 \forall i \in \{1 \dots N\}
 \end{aligned} \tag{39}$$

where

$$\begin{aligned}
 b_i &= \frac{\mu N \sigma_o^2}{2} \left(1 - (1-\mu\sigma_x^2)^2 + \mu^2 (1-\alpha^2) \sigma_{xd}^2 \right) \\
 c_i &= \frac{\mu N \sigma_o^2}{2} \mu^2 (1-\alpha^2) \sigma_{xd}^2
 \end{aligned} \tag{40}$$

Given the stability condition of (9), it follows that $\mu\sigma_x^2 < 1$, for $N > 2$.

Thus $(1-\mu\sigma_x^2) < 1$ and therefore $b_i > 0$. Similarly, it is also noted that $c_i > 0$. Quadratic solutions of (39) are given by (41)

$$\begin{aligned}
 p_i^n &= \frac{1}{2} \left[b_i - \sqrt{b_i^2 - 4c_i} \right] & p_i^p &= \frac{1}{2} \left[b_i + \sqrt{b_i^2 - 4c_i} \right] \\
 p_i^n &\leq p_i^p
 \end{aligned} \tag{41}$$

Furthermore, noting that $b_i \geq \sqrt{b_i^2 - 4c_i}$, for all real solutions, from (41), (42) follows.

$$p_i^n \leq p_i^p \leq b_i \tag{42}$$

Given that the first factor of b_i in (40) is the same MSD bound as LMS, for a bound lower than LMS to hold, the second factor of (40) should be less than unity. Equation (43) identifies the requirement for this assertion.

$$\begin{aligned}
& 1 - (1 - \mu\sigma_x^2)^2 + \mu^2(1 - \alpha^2)\sigma_{xd}^2 < 1 \\
& - (1 - \mu\sigma_x^2)^2 + \mu^2(1 - \alpha^2)\sigma_{xd}^2 < 0 \\
& \sigma_{xd}^2 < \frac{(1 - \mu\sigma_x^2)^2}{\mu^2(1 - \alpha^2)} < \frac{\left(1 - \frac{2\sigma_x^2}{N\sigma_x^2}\right)^2}{\mu^2(1 - \alpha^2)} = \frac{\left(1 - \frac{2}{N}\right)^2}{\mu^2(1 - \alpha^2)}
\end{aligned} \tag{43}$$

Based on (44), this requirement may be expressed in terms of observation noise, (45), where the last line of (44) has been obtained under the assumption of white noise input and independence of weights. Since both μ and α are independent design parameters, they may be chosen in such manner to unconditionally satisfy any range of σ_o^2 , e.g. for α close to unity, the allowable range is ∞ .

$$\begin{aligned}
\sigma_{xd}^2 &= E[Xdd^T X^T] \\
&= E\left[XX^T(W^{Opt} + O)(W^{Opt} + O)^T XX^T\right]
\end{aligned} \tag{44}$$

$$\begin{aligned}
&= R_{ww} = \delta_{ij} w_i w_j \begin{cases} w_j^2 + \sigma_o^2 & j = i \\ \sigma_o^2 & j \neq i \end{cases} \\
&\sigma_o^2 < \frac{\left(1 - \frac{2}{N}\right)^2}{\mu^2(1 - \alpha^2)} - \max(w_i^2)_{i \in \{1 \dots N\}}
\end{aligned} \tag{45}$$

Hence, $\left\{ \begin{array}{l} \exists \alpha \text{ and } \mu \mid p_i^n \leq p_i^P \leq b \leq LMS_{MSD} \\ \forall i \in \{1 \dots N\} \end{array} \right.$, that is, an MSD bound, lower than LMS bound is possible.

5. Partial Update Kalman Least Mean Square Algorithms for System Tracking

This section presents a novel partial update algorithm for FIR adaptive filters based on a Kalman background engine. In the proposed system, a Kalman filter is setup with the coefficients of the full adaptive filter being the states to be estimated based on the observation of a subset of filter coefficients being updated. It is shown that this setup allows for an improved estimation of the full set of filter coefficients, despite the partial update. We propose two methods for postmortem improvements on an ordinary M-Tap periodic update LMS. We also propose a Kalman feedback method, in conjunction with a 1-Tap periodic update LMS, which has a similar performance as a full length LMS, for non-stationary system identification.

5.1 Length Constrained Adaptive FIR Kalman Filter (LCAFKF)

Let the ideal, vector form, full length LMS update be described as in (46).

$$w_{k+1} = w_k + \mu_k x_k e_k \tag{46}$$

where w_k is the $N \times 1$ weight vector. The length-constrained adaptive filter, $M < N$, can be described as (47)

$$\mathbf{w}_k^O = \mathbf{D}_k \mathbf{w}_k \quad (47)$$

where $\mathbf{w}_k^O$ is the observable length-constrained weight vector of length $M \times 1$, $M \leq N$, and $\mathbf{D}_k$, $M \times N$, is the matrix determining which coefficients of $\mathbf{w}_k$ are present in $\mathbf{w}_k^O$ and is given by:

$$\mathbf{D}_k = [\mathbf{I} | \mathbf{0}] \quad (48)$$

where $\mathbf{I}$ is the $M \times M$ identity matrix and $\mathbf{0}$ is $M \times (N-M)$ all-zero matrix. Our objective is to estimate $\mathbf{w}_k$ based on the limited knowledge obtained from $\mathbf{w}_k^O$. Considering the above definitions, we now propose to formulate the problem as a Kalman filtering system. A typical Kalman filter is described as in (49):

$$\begin{aligned} \mathbf{w}_{k+1} &= \mathbf{w}_k + \mathbf{x}_k \xi_k \\ \mathbf{w}_k^O &= \mathbf{D}_k \mathbf{w}_k \end{aligned} \quad (49),$$

where ξ_k is a Gaussian noise with variance $\mu^2 J(\infty)$.

With the formulation complete, we now consider the feasibility of the solution to this problem. The observability matrix of this Kalman filter described above is given by:

$$\mathbf{O} = [\mathbf{D}_k \mathbf{I} \dots \mathbf{D}_k \mathbf{I}^{N-1}]^T \quad (50)$$

However, we can see that the rank of $\mathbf{O}$ in (50) is $M < N$. It then follows that with the current Kalman formulation, the optimal estimate of the full weight vector $\mathbf{w}_k$ cannot be obtained, based on observations of a length-constrained adaptive filter $\mathbf{w}_k^O$. This result is intuitive and the details are included here as an introduction for the next system.

5.2 Partial Update Adaptive FIR Kalaman Filter (PUAFKF)

Given that the original objective is to estimate the full length coefficient vector using fewer observations, we now consider other forms of observation matrix, along with their accompanying assumptions. Of particular interest is the partial coefficient update case, where the observation vector is indeed full length $N \times 1$, but the observation is limited by the fact that only a subset of M coefficients is periodically updated per iteration. That is, in the first time step, only the first M taps are updated, in the second time, taps $M+1$ through $2M$ are updated and so on. The updated coefficients will be accurately observed with zero observation error while the remaining coefficients that were not updated will have an observation error equivalent to the missing update term with negative sign. The observed coefficients $\mathbf{w}_k^O$ can then be expressed as:

$$\mathbf{w}_k^O = \mathbf{w}_k - \begin{bmatrix} \mathbf{0} \\ \sum_{i=0}^{\left\lceil \frac{N}{M} \right\rceil} \mu_k e(k-i) x(k-i-M) \\ \vdots \\ \sum_{i=0}^m \mu_k e(k-i) x(k-i-2M) \\ \vdots \\ \mu_k e(k) x(k-N-M) \\ \vdots \\ \mu_k e(k) x(k-N-1) \end{bmatrix} \quad (51)$$

where $\mathbf{0}$ is an $M \times 1$ zero vector, and $m = \left\lceil \frac{N}{M} \right\rceil$ is the number of M -blocks in an N tap filter.

Assuming LMS update and grouping the error terms together, with (51) can be rewritten as:

$$\mathbf{w}_k^O = \mathbf{w}_k - \mu \left\{ \begin{bmatrix} \mathbf{0} \\ e(k) x(k-M) \\ \vdots \\ x(k-N-1) \end{bmatrix} + \dots + \begin{bmatrix} \mathbf{0} \\ \vdots \\ x(k-m-M) \\ \vdots \\ x(k-m-2M) \\ \mathbf{0} \end{bmatrix} \right\} \quad (52)$$

Comparing (52) with the observation equation in Eq (49), it follows that the observation matrix for this partial update case is an identity matrix of size $N \times N$. Considering the observability matrix, (19), it is clear that O is of rank N , *i.e.* the system is observable and we can indeed get an estimate for the full length filter coefficients based on a partially updated set of coefficients. To complete the necessary formulation, a stochastic description of observation noise and update terms in (52) is required. Proceeding under the typical LMS setup assumption, the problem simplifies to finding the mean and correlation matrix of the $e(k)x(k-i)$ process, for $i \in \{1, \dots, N-1\}$. The mean of this process is estimated as in (53).

$$\begin{aligned} E[e(k)x(k-i)] &= E[(d(k) - \mathbf{x}_k^T \mathbf{w}_k) x(k-i)] \\ &= E[d(k)x(k-i)] - w_k^i \sigma_x^2 \\ &= w_i^W \sigma_x^2 - w_k^i \sigma_x^2 \\ &= \Delta w_k^i \sigma_x^2 \end{aligned} \quad (53)$$

Where $\mathbf{w}^W$ is the optimal solution, and the third line follows from Wiener-Hopf equation and the white Gaussian assumption of the input. Noting that

$\Delta w_{k+1}^i = (1 - \mu\sigma_x^2)\Delta w_k^i - \mu x(k-i)e^O(k)$, (Haykin, 2002), where e^O is the additive observation noise, then (53) simplifies to (54)

$$\Delta w_k^i r_0^x = \left[-\mu e_k^O x(k-i) * \left(1 - \mu\sigma_x^2\right)^k U(k) \right] \sigma_x^2 \quad (54)$$

$$E[e(k)x(k-i)] = \Delta w_k^i \sigma_x^2 \xrightarrow{k \rightarrow \infty} 0$$

where $U(\cdot)$ is the step function, and $*$ the convolution operation. The second line of (54) follows from the exponential decay of the convolution term. The variance of this process is estimated as:

$$\begin{aligned} E[e(k)x(k-i)e(k)x(k-j)] &= \\ E[e(k)e(k)]E[x(k-i)x(k-j)] &+ \\ + E[e(k)x(k-i)]E[e(k)x(k-j)] & \\ = J(\infty)\sigma_x^2 & \end{aligned} \quad (55)$$

The second line follows from the Gaussian moment factoring theorem, (Haykin, 2002), and the third line from (54).

5.3 Performance Evaluation through Simulations

The proposed system described above is composed of a regular adaptive FIR filter of length N , where M coefficients are being updated sequentially, at every iteration. The observed coefficient vector is fed into a Kalman filter as the observation vector to produce an improved estimate for the full length adaptive filter. We consider two scenarios. The first scenario, the Standard Algorithm (SA), the partial updates FIR are fed into the Kalman engine which provides the estimate for the full length filter coefficients in a postmortem fashion. This corresponds to a case where a limited complexity remote system does the preliminary work and then feeds the result to the Kalman filter for improved estimation. In the second scenario, LMS Kalman Feedback Algorithm (LKFA), the output of the adaptive filter is fed to the Kalman filter, as before, furthermore, the Kalman filter output is then fed back to update all the FIR filter coefficients before the next iteration.

To verify the accuracy of the assumptions leading to (55), we estimate the statistics of the observation noise on line as in (56). We refer to this method as Online Observation Plant Noise Variance Estimation Algorithm (OPNVEA).

$$r_i^{ex}(k) = r_i^{ex}(k-1) + (1-\gamma)[e(k)x(k-i)]^2 \quad i \in \{1 \dots N\} \quad 0 < \gamma < 1 \quad (56)$$

We now provide the simulation results for SA, LKFA and OPNVEA methods, under time invariant, slowly varying and finally rapidly varying system tracking. As points of reference, the MSE of a full length LMS filter and 1-Tap periodic update LMS, *i.e.* $M=1$, are also provided. The unknown system has 50 taps. Its first 3 significant coefficients and the accompanying SA identification are depicted in Fig. 4, Fig. 5 and Fig. 6 through Fig. 8. The input variance is 3, observation noise variance is 1, and the full LMS step size is 8×10^{-5} . For the time varying systems of Fig. 5 through Fig. 8, each coefficient is modulated by a sine wave, with a randomly generated frequency of variances 0.5 and 15, respectively. The coefficients of the unknown system are also randomly generated, with amplitude variance of 10.

Results of the simulations are summarized in Table 2. The first column of the Table 2 refers to the time invariant system identification, the next column to the slowly varying system and the final column to the rapidly varying system. The step size for the 1-Tap periodic update LMS ($M=1$) is 4×10^{-3} , the forgetting factor γ for OPNVEA (Scenario 1) is 0.1. As seen in Table 2, all proposed algorithms lead to improvements, compared to 1-Tap periodic LMS algorithm. These improvements become more pronounced in non-stationary system estimation. It is further noted that the performance of the OPNVE is similar to SA, even in the non-stationary case, confirming the validity of the assumptions leading to (55). It is particularly worthwhile to note that the performance, in non-stationary cases, of the full LMS and 1-Tap LKFA are quite similar. With respect to the temporal complexity of the LKFA, it should be noted that since the correlation matrices of the system and observation plants are known ahead of time, the most time consuming portions of the Kalman filter, *i.e.* the post covariance matrix and Kalman gain, can be calculated off-line, thus significantly reducing the real-time constraints on the LKFA.

Fig. 4. Stationary System Identification

Fig. 5. Slowly Varying System Identification

IntechOpen

Fig. 6. Rapidly Varying System Identification, 1st Coefficient

Fig. 7. Rapidly Varying System Identification, 2nd Coefficient

Fig. 8. Rapidly Varying System Identification, 3rd Coefficient

Method Name	Stationary MSE [10 ³]	Slowly Varying MSE [10 ⁴]	Rapidly Varying MSE [10 ⁴]
Full LMS	0.003	0.035	1.607
1 Tap LMS	9.915	2.956	36.39
OPNVEA	9.373	2.788	34.40
SA	9.501	2.832	34.83
LKFA	10.217	1.274	2.357

Table 2. Summary of Periodic Update Results

6. Conclusion and future work

We have shown the applicability and feasibility of the Kalman filter modeling for LMS-based algorithms. Particularly, it was observed that such modeling allows for realizations of update methods with MSD bounds lower than those obtained by LMS. Furthermore, it was

shown that the system and observation plant setup provides for a convenient formulation of a length constraint adaptive FIR filters. In particular, it was shown that this setup allows for both post mortem and real-time optimization of system identification process. The future work would focus on application of the current method on update policies, other than the periodic one. Furthermore, the application of smoothing Kalman filter on length-constrained MSE would be investigated.

It is noted that the improvements gained by applying the standard Kalman filter are temporally localized. This is indeed in line with the linear structure of a typical Kalman filter. As such the application of the Extended Kalman filter will also be investigated under non-stationary environments.

7. References

- Amit Kumar Kohli , D. K. Mehra (2008), Adaptive Multiuser Channel Estimation using Reduced Kalman/LMS Algorithm, [Wireless Personal Communications](#), Vol. 64, No. 4, 507-521, 0929-6212
- Christian Bauckhage and John K. Tsotsos(2005). Separable Linear Classifiers for Online Learning in Appearance Based Object Detection, In: [Computer Analysis of Images and Patterns](#), A. Gagalowicz and W. Philips , 347-353, Springer, 978-3-540-28969-2 , Berlin
- Frank L. Lewis,(1986)*Optimal Estimation With An Introduction to Stochastic Control Theory*, John Wiley and Sons, 0-47183741-5, New York
- John R. Taylor, (1997) *An Introduction to Error Analysis*, University Science Books, 0-935702-42-3, Sausalito
- K. Mayyas and T. Aboulnasr (2004), Reduced-Complexity Transform-Domain Adaptive Algorithm with Selective Coefficient Update, *IEEE Trans. On Cir. and Sys. II*, Vol. 51, No. 3, 136-142
- Mal-Rey Lee (2001), 3D Shape Reconstruction of Hybrid Reflectance Using the LMS Algorithm. [IJPRAI](#), Vol. 15, No. 4, 723-734, 1793-6381
- Masayuki Hoshino, Takeo Ohgane, and Yasutaka Ogawa, Adaptive DEF Control with Differential LMS Algorithm, *Electronics and communication in Japan - Part 1*, Vol. 84, No. 1, Nov. 2001, 99-108, 87566621
- Paulo A. C. Lopes, Gonçalo Tavares and José B. Gerald, A New Type of Normalized LMS Algorithm Based on the Kalman Filter, *ICSAAP 2007*, Vol. 3, 1345-1348, 1-4244-0727-3, USA, April 2007, Honolulu
- S.C. Douglas (1997), Adaptive Filters Employing Partial Updates, *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 44, No. 3, 209-216
- Simon Haykin, (2002) *Adaptive Filter Theory*, 4th Ed., Prentice Hall, 0-13-048434-2, New Jersey
- Sristi, P. Lu, W.-S. and Antonion, A. (2001), [ISCAS 2001](#), pp. 721-724, 0-7803-6685-9, Australia, May, 2001, IEEE, Sydney
- Yang Xinyu, Zeng Ming, Zhao Rui and Shi Yi(2004). A Novel LMS Method for Real-Time Network Traffic Prediction, *Proceedings of ICCSA 2004*, 12-136, 978-3-540-22060-2, Italy, May 2004, Springer, Assisi

IntechOpen

IntechOpen

Multimedia

Edited by Kazuki Nishi

ISBN 978-953-7619-87-9

Hard cover, 452 pages

Publisher InTech

Published online 01, February, 2010

Published in print edition February, 2010

Multimedia technology will play a dominant role during the 21st century and beyond, continuously changing the world. It has been embedded in every electronic system: PC, TV, audio, mobile phone, internet application, medical electronics, traffic control, building management, financial trading, plant monitoring and other various man-machine interfaces. It improves the user satisfaction and the operational safety. It can be said that no electronic systems will be possible without multimedia technology. The aim of the book is to present the state-of-the-art research, development, and implementations of multimedia systems, technologies, and applications. All chapters represent contributions from the top researchers in this field and will serve as a valuable tool for professionals in this interdisciplinary field.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

NastooH Avesta (2010). KlamAn Assisted LMS Methods, Multimedia, Kazuki Nishi (Ed.), ISBN: 978-953-7619-87-9, InTech, Available from: <http://www.intechopen.com/books/multimedia/klaman-assisted-lms-methods>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen