Proc. Biennial Symp. Commun. (Kingston, ON), pp. 164-167, June 1990

JOINT TIME DELAY ESTIMATION AND ADAPTIVE RECURSIVE LEAST SQUARES FILTERING

Daniel Boudreau [1,3] and Peter Kabal [1,2]

[1] Electrical Engineering
McGill University
Montreal, Quebec, H3A 2A7

[2] INRS-Télécommunications
Université du Québec
Verdun, Quebec, H3E 1H6

[3] Communications Research Centre
Department of Communications
Ottawa, Ontario, K2H 8S2

Abstract

A general estimation model is defined in which two observations are available; one being a noisy version of the transmitted signal, while the other is a noisy filtered and delayed version of the same transmitted signal. Examples of such systems occur in noise or echo cancellation, digital communication or geophysical exploration. The delay and the filter are unknown quantities that must be estimated. An adaptive system, based on the least squares (LS) estimation criterion, is proposed in order to perform a joint estimation of the two unknowns. The joint estimator is conceptually composed of an adaptive delay element operating in conjunction with a transversal adaptive filter. The weighted sum of squared errors is minimized with respect to the delay and the adaptive filter weight vector. The latter is adapted using a fast version of the recursive least squares (RLS) algorithm, while the former is updated using a form of derivative, with respect to the delay, of the sum of squared errors. In order to perform this task efficiently, the adaptive delay is limited to integer values and is corrected one sample at a time. The integer delay value is defined as the lag. A series of relations is presented, in order to compute and update the lag value such that the optimum least squares solution is attained. The joint delay estimation and RLS filtering algorithm is obtained by combining the lag update relations with a version of the fast transversal filter RLS algorithm. The simulations of the resulting algorithm show that both stationary and time-varying delays are effectively tracked and that the adaptive filter estimates properly the reference filter impulse response.

1. Introduction

Estimating the time delay between two versions of the *same* signal, each one corrupted by zero-mean uncorrelated noise components, has been the subject of much research effort over recent years. In this paper, we consider the more difficult model in which the delayed path is subject to frequency dependent attenuation. The corresponding discrete-time model is of the form

$$\begin{aligned} y_1(n) &= s(nT) + v_1(nT) \\ y_2(n) &= \mathcal{L}_{D_n,h(n)}[s(nT)] + v_2(nT), \end{aligned} \qquad (1)$$

where T is the sampling period, $s(t)$ is the transmitted signal, D_n is a time-varying delay, $v_1(t)$ and $v_2(t)$ are zero-mean noise processes, uncorrelated with each other, as well as with $s(nT)$, and $\mathcal{L}_{D_n,h(n)}[s(nT)]$ is an *unknown* linear operator that filters a delayed version of $s(n)$.

It is of interest to estimate $\mathcal{L}_{D_n,h(n)}[s(nT)]$ or its inverse when the operator takes the form

$$\mathcal{L}_{D_n,h(n)}[s(nT)] = h(n) \otimes s(nT - D_n) \qquad \text{Type I}, \qquad (2)$$

corresponding to the filtering of a delayed version of $s(n)$, or

$$\mathcal{L}_{D_n,h(n)}[s(nT)] = h(t) \otimes s(t)|_{t=nT-D_n} \qquad \text{Type II}, \qquad (3)$$

corresponding to a filter followed by a delay. Note that the operator $\otimes$ denotes the convolution operator and that the two equations define different types of systems, as indicated.

In order to estimate $\mathcal{L}_{D_n,h(n)}[s(nT)]$ or its inverse, we propose to use a joint estimator that is composed, at least conceptually, of an *adaptive delay element* d_n and a conventional M^{th}-order adaptive FIR filter with weight vector $\mathbf{w}_M(n)$. Depending on the system configuration, the adaptive delay element is in series or in parallel with the adaptive filter. This new joint adaptive configuration is based on the use of the same error signal for the adaptation of both systems. In this paper, we consider a Type II system that takes the form of Fig. 1.

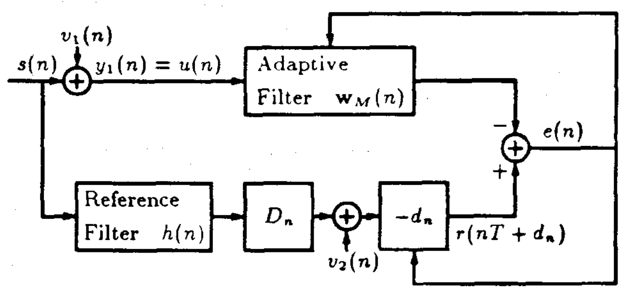

Fig. 1 Identification of a Type II system

Adaptive delay elements have been studied in recent work [1], [2] for the simple signal model where the filter $h(n)$ is absent and in [3] for the signal model of equation (1), when both the delay estimation and the filter estimation are performed by minimizing the mean squared error (MSE) $\xi_n = E[|e(n)|^2]$ with the help of gradient-based techniques (steepest-descent and LMS).

A simple adaptive filter has the potential to model the functional $\mathcal{L}_{D_n,h(n)}[s(nT)]$ since this function can be approximated by an FIR filter with the proper number of taps. But this approach is inefficient in the sense that the reference delay D_n is modelled by a shift in the adaptive filter impulse response. For a fixed filter order M, this shift may result into an error that is larger than the error corresponding to perfect modelling. An additional adaptive delay estimation algorithm, specifically designed to track the reference delay variations, allows a better impulse response centering and the use of an adaptive filter with a smaller order. We refer to a "centered" adaptive filter impulse response as one corresponding to the lowest modelling error.

2. Least Squares Estimation with a Delay Element

In this paper, we present a new algorithm for the joint minimization, with respect to the delay value d_n and the adaptive filter weight vector $\mathbf{w}_M(n)$, of the sum of error squares

$$E_M(n) = \sum_{i=1}^{n} \lambda^{n-i}|e_M(i,d_i)|^2, \qquad (4)$$

where M is the filter order and λ is a positive memory factor between zero and one. The joint delay and reference filter tracking algorithm that we propose is based on the assumption that a Type II adaptive system is used (the adaptive delay is located in the reference branch) and that the delay corrections are done one sample at a time. Define this integer time delay as a time *lag* and denote it by ℓ. Then, the error $e_M(i,d_i)$ in equation (4) can be expressed as

$$e_M(i,d_i) = e_M(i,\ell) = r(i+\ell) - \mathbf{w}_M^H(n)\mathbf{u}_M(i), \qquad (5)$$

where H denotes complex conjugate transpose, $r(n)$ is the output of the reference branch, $u(n)$ is the input to the adaptive branch (signal $y_1(n)$ in Fig. 1) and the vector $\mathbf{u}_M(n)$ is the vector of M input samples in the adaptive filter delay line.

For a given value of ℓ, the minimum sum of weighted error squares $\hat{\xi}_M(n,\ell)$ is defined as

$$\hat{\xi}_M(n,\ell) = \min_{\mathbf{w}_M(n)} E_M(n). \qquad (6)$$

The weight vector for which this minimum is attained is defined as $\hat{\mathbf{w}}_M^\ell(n)$. If the adaptive delay d_i is not equal to D_i, for all i, the sum of errors $\hat{\xi}_M(n,d_n)$ is not minimum with respect to d_n, unless the adaptive filter length is large enough to accommodate both the modelling of the reference filter $h(n)$ and the reference delay (i.e. M is large enough such that the delayed optimum adaptive weight vector is not truncated).

3. Lag-recursive Relations

Our joint LS lag estimation and adaptive filtering algorithm can be cast into the following general algorithmic form

1. Apply the RLS algorithm in order to obtain $\hat{\mathbf{w}}_M^\ell(n)$ and $\hat{\xi}_M(n,\ell)$
2. Adapt ℓ by using derivative information from $\hat{\xi}_M(n,\ell)$.

Conceptually, the first part of the algorithm can be implemented by using any of the computationally efficient forms of the RLS algorithm, and the second part can be implemented as a gradient search, with respect to ℓ, of $\hat{\xi}_M(n,\ell)$. The gradient is given by

$$\frac{\partial \hat{\xi}_M(n,\ell)}{\partial \ell} = \begin{cases} 1 & \text{if } \begin{cases} \hat{\xi}_M(n,\ell+1) < \hat{\xi}_M(n,\ell) & \text{and} \\ \hat{\xi}_M(n,\ell+1) < \hat{\xi}_M(n,\ell-1) \end{cases} \\ -1 & \text{if } \begin{cases} \hat{\xi}_M(n,\ell-1) < \hat{\xi}_M(n,\ell) & \text{and} \\ \hat{\xi}_M(n,\ell-1) < \hat{\xi}_M(n,\ell+1) \end{cases} \\ 0 & \text{otherwise,} \end{cases} \qquad (7)$$

and the lag value is updated as

$$\ell = \ell + \left\langle \frac{\partial \hat{\xi}(n,\ell)}{\partial \ell} \right\rangle, \qquad (8)$$

where $\langle\cdot\rangle$ denotes a time average. Note that when ℓ is updated, $\hat{\mathbf{w}}_M^\ell(n)$ must also be corrected, in order to obtain the joint solution of (6).

In order to compute (7), the optimum weight vectors for lags $\ell+1$ and $\ell-1$ must be available. This implies that two parallel applications of the RLS algorithm are necessary, in addition to the one generating $\hat{\mathbf{w}}_M^\ell(n)$. This also implies some problems when the value of ℓ gets updated in one direction or the other, since a new parallel branch must be initialized with the optimum weight vector corresponding to the new value of the lag. But this new optimum weight vector depends on past data (at least within the memory of the algorithm) and is unknown.

We solve this transition problem by deriving a set of lag-recursive relations that allow the *exact* computation of $\hat{\xi}_M(n,\ell+1)$, $\hat{\xi}_M(n,\ell-1)$, $\hat{\mathbf{w}}_M^{\ell+1}(n)$ and $\hat{\mathbf{w}}_M^{\ell-1}(n)$ from the knowledge of $\hat{\mathbf{w}}_M^\ell(n)$ and $\hat{\xi}_M(n,\ell)$. These lag-recursive relationships involve some variables that are defined in the following list:

- $\mathbf{a}_{M-1}(n)$ and $\mathbf{b}_{M-1}(n)$ are the optimum weight vectors of the forward and backward linear predictors of order $M-1$ for the input signal $u(n)$
- $F_{M-1}(n)$ and $B_{M-1}(n)$ are the corresponding minimum values of the sums of weighted forward and backward prediction error squared, $f_{M-1}(n)$ and $b_{M-1}(n)$
- $v_{M-1}^{f\ell}(n)$ and $v_{M-1}^{b\ell}(n)$ are the correlation functions between the $(M-1)^{th}$ order forward and backward prediction error sequences, $\{f_{M-1}(n)\}$ and $\{b_{M-1}(n)\}$ respectively, and the desired response at lag ℓ, i.e.

$$v_{M-1}^{f\ell}(n) = \sum_{i=1}^{n} \lambda^{n-i} f_{M-1}(i) r^*(i+\ell)$$

$$v_{M-1}^{b\ell}(n) = \sum_{i=1}^{n} \lambda^{n-i} b_{M-1}(i) r^*(i+\ell)$$

  where $*$ denotes complex conjugate.
- $\alpha_{M-1}(n,\ell)$ is the a priori estimation error defined as

$$\alpha_{M-1}(n,\ell) = r(n+\ell) - \mathbf{w}_{M-1}^{\ell H}(n-1)\mathbf{u}_{M-1}(n)$$

- $\mathbf{g}_{M-1}(n)$ is the Kalman gain of order $M-1$ appearing in the RLS algorithm
- $\lceil \hat{\mathbf{w}}_M^\ell(n) \rceil_{M-1}$ is defined as the $(M-1)$-vector corresponding to the *first* components of $\hat{\mathbf{w}}_M^\ell(n)$
- $\lfloor \hat{\mathbf{w}}_M^\ell(n) \rfloor_{M-1}$ is the $(M-1)$-vector corresponding to the *last* components of $\hat{\mathbf{w}}_M^\ell(n)$
- $w_{1M}^\ell(n)$ and $w_{MM}^\ell(n)$ are respectively the first and last component of $\hat{\mathbf{w}}_M^\ell(n)$

The the lag-recursive relationships are:

<u>$\hat{\xi}_M(n,\ell) \rightarrow \hat{\xi}_M(n,\ell+1)$</u>

$$\begin{aligned} \hat{\xi}_{M-1}(n-1,\ell+1) &= \hat{\xi}_M(n,\ell) + \frac{|v_{M-1}^{f\ell}(n)|^2}{F_{M-1}(n)} \\ \hat{\xi}_{M-1}(n,\ell+1) &= \lambda\hat{\xi}_{M-1}(n-1,\ell+1) \\ &\quad + \alpha_{M-1}^*(n,\ell+1)e_{M-1}(n,\ell+1) \\ \hat{\xi}_M(n,\ell+1) &= \hat{\xi}_{M-1}(n,\ell+1) - \frac{|v_{M-1}^{b(\ell+1)}(n)|^2}{B_{M-1}(n)} \end{aligned} \qquad (9)$$

<u>$\hat{\xi}_M(n,\ell) \rightarrow \hat{\xi}_M(n,\ell-1)$</u>

$$\begin{aligned} \hat{\xi}_{M-1}(n,\ell) &= \hat{\xi}_M(n,\ell) + \frac{|v_{M-1}^{b\ell}(n)|^2}{B_{M-1}(n)} \\ \hat{\xi}_{M-1}(n-1,\ell) &= \lambda^{-1}\hat{\xi}_{M-1}(n,\ell) - \lambda^{-1}\alpha_{M-1}^*(n,\ell)e_{M-1}(n,\ell) \\ \hat{\xi}_M(n,\ell-1) &= \hat{\xi}_{M-1}(n-1,\ell) - \frac{|v_{M-1}^{f(\ell-1)}(n)|^2}{F_{M-1}(n)}. \end{aligned} \qquad (10)$$

<u>$\hat{\mathbf{w}}_M^\ell(n) \rightarrow \hat{\mathbf{w}}_M^{\ell+1}(n)$</u>

$$\begin{aligned} \hat{\mathbf{w}}_{M-1}^{\ell+1}(n-1) &= \lfloor \hat{\mathbf{w}}_M^\ell(n) \rfloor_{M-1} + \mathbf{a}_{M-1}(n)w_{1M}^\ell(n) \\ \hat{\mathbf{w}}_{M-1}^{\ell+1}(n) &= \hat{\mathbf{w}}_{M-1}^{\ell+1}(n-1) - \lambda^{-1}\mathbf{g}_{M-1}(n)e_{M-1}(n,\ell+1) \\ \hat{\mathbf{w}}_M^{\ell+1}(n) &= \begin{bmatrix} \hat{\mathbf{w}}_{M-1}^{\ell+1}(n) \\ 0 \end{bmatrix} + \frac{v_{M-1}^{b(\ell+1)}(n)}{B_{M-1}(n)} \begin{bmatrix} -\mathbf{b}_{M-1}(n) \\ 1 \end{bmatrix}. \end{aligned} \qquad (11)$$

$\underline{\hat{\mathbf{w}}_M^{\ell}(n) \rightarrow \hat{\mathbf{w}}_M^{\ell-1}(n)}$

$$\hat{\mathbf{w}}_{M-1}^{\ell}(n) = \lceil \hat{\mathbf{w}}_M^{\ell}(n) \rceil_{M-1} + \mathbf{b}_{M-1}(n) w_{MM}^{\ell}(n)$$

$$\hat{\mathbf{w}}_M^{\ell-1}(n) = \begin{bmatrix} 0 \\ \hat{\mathbf{w}}_{M-1}^{\ell}(n-1) \end{bmatrix} + \frac{v_{M-1}^{f(\ell-1)}(n)}{F_{M-1}(n)} \begin{bmatrix} 1 \\ -\mathbf{a}_{M-1}(n) \end{bmatrix}. \tag{12}$$

We refer to equations (9) and (11) as the *forward lag update recursions*, and to equations (10) and (12) as the *backward lag update recursions.*

The variables used in the above relationships, except for $v_{M-1}^{f\ell}(n)$ and $v_{M-1}^{b\ell}(n)$, are intermediate variables available in the implementation of the fast transversal filter (FTF) RLS algorithm [4]. As for $v_{M-1}^{f\ell}(n)$ and $v_{M-1}^{b\ell}(n)$, they are found in the implementation of the recursive least-squares lattice filter [5]. This close relationship between the above lag-recursive equations and the fast implementations of the RLS algorithm is taken into consideration in the formulation of the joint time delay estimation and RLS adaptive filtering algorithm.

The lag-recursive relationships can be derived by using geometrical arguments or by using shift-invariance properties of some vectors and matrices appearing in the RLS algorithm [6]. These relationships can be combined with a version of the fast transversal filter RLS algorithm in order to obtain a joint time delay estimation and RLS filtering algorithm.

4. Fast Joint Time Delay Estimation and Adaptive Filtering RLS Algorithms

Based on the error and weight vector recursions developed in the previous section, joint time delay and FTF algorithms can be obtained. These algorithms are composed of three distinct computational phases. The first phase is essentially the preliminary computations phase of the FTF algorithm [4], with a slight difference. This difference resides in the order of the forward and backward predictors, which must be $M-1$ in the present case. The second computational phase involves the computation of the current weight vector $\hat{\mathbf{w}}_M^{\ell}(n)$ and the computation of the three errors $\hat{\xi}_M(n,\ell)$, $\hat{\xi}_M(n,\ell+1)$ and $\hat{\xi}_M(n,\ell-1)$. These computations are performed by using the lag update recursions for the error and the weight vector. It first involves the computation of $\hat{\mathbf{w}}_M^{\ell-1}(n)$ and $\hat{\xi}_M(n,\ell-1)$. Then the forward lag recursions, for both the error and the weight vector, are used twice in order to get the errors for ℓ and $\ell+1$ and the weight vector for ℓ. These successive applications of the forward lag recursions produce the least number of computations, compared for example to the application of the forward and backward lag recursions on the error and weight vector at lag ℓ. This choice also simplifies the third computational phase, which involves a decision on the lag update and the computations of the new corresponding variables.

Schematically, the first two phases of the algorithm can be represented as in Fig. 2, where six parallel digital filters are represented. The top three filters are essentially the same as the ones used in the conventional fast transversal filter [4], [5], except for the difference in predictor order, and represent the first computational phase. Note that $\eta_{M-1}(n)$ and $\psi_{M-1}(n)$ are the *a priori* forward and backward prediction errors, and $\gamma_M(n)$ is defined as

$$\gamma_M(n) = 1 + \lambda^{-1}\mathbf{g}_M^H(n)\mathbf{u}_M(n).$$

The fourth filter is for the computation of $\hat{\xi}_M(n,\ell-1)$ and $\hat{\mathbf{w}}_M^{\ell-1}(n-1)$. Notice that $\hat{\xi}_{(M-1)}(n-1,\ell)$ is also obtained from that filter, through the first relation of (9). A fifth filter, with weight vector $\hat{\mathbf{w}}_{M-1}^{\ell}(n-1)$, is used to obtain $v_{M-1}^{b\ell}(n)$, from which $\hat{\xi}_M(n,\ell)$ and $\hat{\xi}_{(M-1)}(n,\ell+1)$ are computed. Finally a sixth transversal filter, with weight vector $\hat{\mathbf{w}}_{M-1}^{\ell+1}(n-1)$, is used in the computation of $v_{M-1}^{b(\ell+1)}(n)$ and $\hat{\xi}_M(n,\ell+1)$.

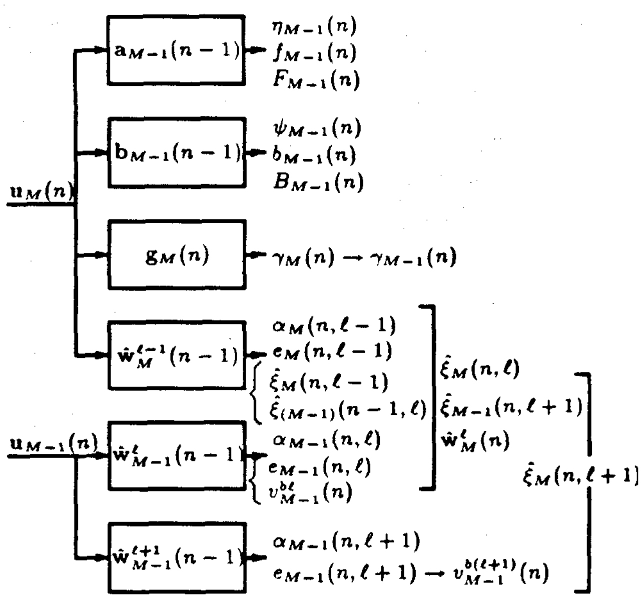

Fig. 2 Interpretation of the computation of the sums of squared errors in terms of transversal filters

The originality of this joint LS algorithm resides in the serial computations, from $\hat{\mathbf{w}}_M^{\ell-1}(n-1)$, of all the necessary errors and weight vectors for lags ℓ and $\ell+1$. The lag-update recursions append themselves nicely to the FTF algorithm of Cioffi [4], with $M-1$-order predictors. Note also that all the computations of the joint algorithm based on Fig. 2 are exact as long as the lag is not updated. When this situation happens, the lag-recursive relations of equations (9) to (12) are also useful in obtaining a smooth transition of $\hat{\mathbf{w}}_M^{\ell+1}(n)$ or $\hat{\mathbf{w}}_M^{\ell-1}(n)$ to their new values, by making use of the past data.

An alternative joint algorithm could take the parallel form of Fig. 3, where three versions of the FTF algorithm are implemented, one for each possible lag, in order to compute $\hat{\xi}_M(n,\ell-1)$, $\hat{\xi}_M(n,\ell)$ and $\hat{\xi}_M(n,\ell+1)$. But this form of joint algorithm does not allow, when the lag is updated at time n_o, the determination of the new sum of squared errors and the new weight vector in the direction of the lag update. These variables have to be initialized to zero. This typically introduces an error in both quantities, because their computation involves the internal variables $\gamma_M(n)$ and $\mathbf{g}_M(n)$, that were obtained from nonzero initial conditions at $n < n_o$. In order to allow a smooth transition in this case, two extra parallel branches, one for $\ell+2$ and one for $\ell-2$, must be computed, which gives a final parallel algorithm involving five branches.

5. Experimental Results

The fast joint time delay estimation and adaptive RLS filtering algorithm corresponding to Fig. 2 was implemented, in order to verify its practical behaviour. The usual problem of numerical instability, often associated with the RLS algorithm implementations [4], was also present in our algorithm. A periodic restart of the algorithm, similar to the technique proposed by Eleftheriou and Falconer [7], was implemented to reinitialize the algorithm's internal variables (those produced by the first computational phase). The restart period was arbitrarily fixed to 600 iterations. The time average of equation (8) was implemented by accumulating, for ℓ, $\ell+1$ and $\ell-1$, the sums of

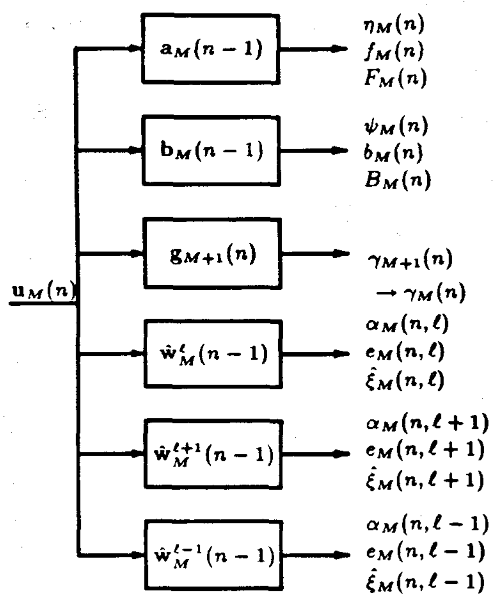

Fig. 3 Parallel application of the FTF algorithm in order to compute $\hat{\xi}_M(n,\ell-1)$, $\hat{\xi}_M(n,\ell)$ and $\hat{\xi}_M(n,\ell+1)$

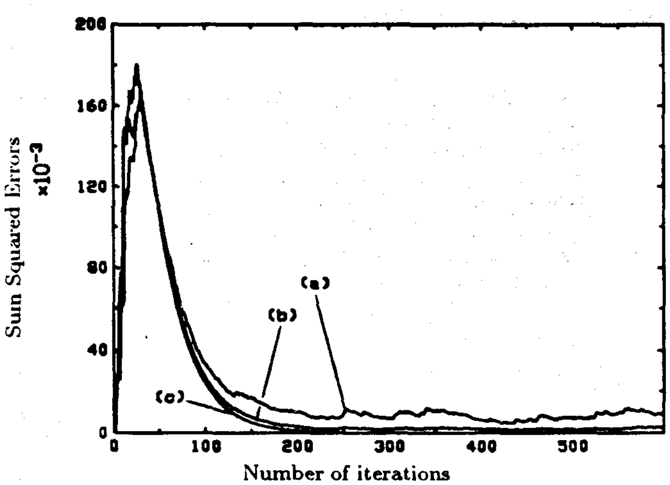

Fig. 4 Variations of the sum of squared errors with ℓ; $\lambda = 0.97$; (a) $\hat{\xi}_M(n,\ell-1)$; (b) $\hat{\xi}_M(n,\ell)$; (c) $\hat{\xi}_M(n,\ell+1)$

squares $\hat{\xi}_M(n,\ell)$ over a number of iterations. The reference filter $h(n)$ was implemented as a 21-tap lowpass FIR filter, with a 3dB bandwidth approximately equal to 0.7π. The transmitted signal $s(n)$ was a zero-mean process with a white power spectral density from $-\pi$ to π. The adaptive filter also had 21 taps.

We were essentially interested in the delay tracking capabilities of the joint LS algorithm. The effect of a reference unit step delay, applied at time zero, on the sums of squared errors, is shown in Fig. 4 when there is no lag update algorithm and $\lambda = 0.97$. After an initial adaptation, the adaptive filter with a delay of $\ell + 1$ (curve (c)) models the reference filter very well, since $\hat{\xi}_M(n,\ell+1)$ is essentially zero. Note that the mean steady-state value of the sum of squares increases for ℓ (curve (b)) and $\ell - 1$ (curve (a)), indicating that the adaptive filter modelling becomes less effective for these delays.

A typical delay estimator response is shown in Fig. 5, when there is no noise and the averaging length, for the derivative estimate, is 26 samples and $\lambda = 0.9$. The reference delay D_n is represented by the dashed line. The joint algorithm follows the reference delay closely, which indicates that the adaptive filter impulse response stays centered, even if the reference delay attains a fairly large value after 1200 iterations. This figure shows the joint algorithm potential in tempering significantly the problem of unwanted delay tracking by the adaptive filter.

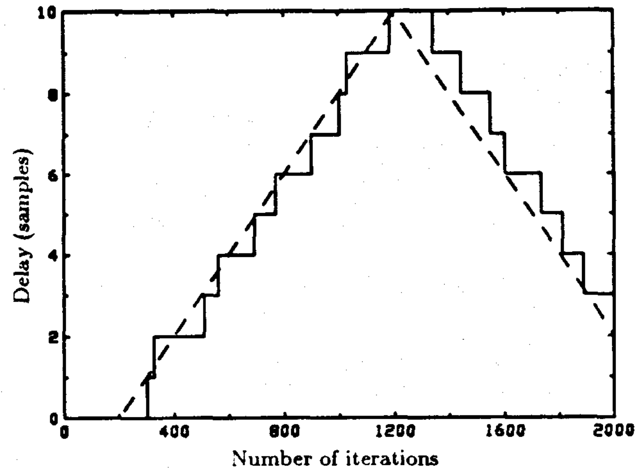

Fig. 5 Tracking of a linearly changing delay; $\lambda = 0.9$

6. Conclusion

This paper has presented a new algorithm for the joint estimation of time delays and correlation function between two observed signals, when the estimation criterion is the minimization of the sum of squared errors. The principal contribution of the work is the derivation of an RLS algorithm that makes use of time update, filter order and lag update relations, in order to compute efficiently the least squared error and the corresponding weight vector. The lag update constitutes an additional degree of freedom for the minimization of the sum of weighted squared errors. It also allows the adaptive filter to model more efficiently the reference filter $h(n)$.

References

1. H. Messer and Y. Bar-Ness, "Closed-loop least mean square time-delay estimator," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 413–424, April 1987.
2. J. O. Smith, B. Friedlander, "Adaptive interpolated time-delay estimation," *IEEE Trans. on Aerospace and Electronic Systems*, vol. AES-21, pp. 180–199, March 1985.
3. D. Boudreau and P. Kabal, "Joint gradient-based time delay estimation and adaptive filtering," *IEEE 1990 International Symposium on Circuits and Systems*, New Orleans, May 1–3, 1990.
4. J. M. Cioffi and T. Kailath, "Fast, recursive-least-squares transversal filters for adaptive filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 304–337, April 1984.
5. S. Haykin, *Adaptive Filter Theory*, Prentice-Hall,1986.
6. N. Kalouptsidis, G. Carayannis and D. G. Manolakis, "Fast design of multichannel FIR least-squares filters with optimum lag," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 48–59, February 1984.
7. E. Eleftheriou and D. D. Falconer, "Tracking properties and steady-state performance of RLS adaptive filter algorithms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 1097–1110, October 1986.