JOINT TIME DELAY ESTIMATION AND ADAPTIVE RECURSIVE LEAST SQUARES FILTERING

Daniel Boudreau¹,² and Peter Kabal¹,²

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 of 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

\[ y_1(n) = s(nT) + v_1(nT) \]
\[ y_2(n) = D_a \ast s(nT) + v_2(nT), \] (1)

where \( T \) is the sampling period, \( s(t) \) is the transmitted signal, \( D_a \) 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 \( D_a \ast s(nT) \) is an unknown linear operator that filters a delayed version of \( s(n) \).

It is of interest to estimate \( D_a \ast s(nT) \) or its inverse when the operator takes the form

\[ D_a \ast s(nT) = h(n) \ast s(nT - D_a) \] (Type I),

(2)

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

\[ L_{D_a \ast s(nT)}[s(nT)] = h(t) \ast s(t) |_{t=nT-D_a} \] (Type II), (3)

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

In order to estimate \( L_{D_a \ast s(nT)}[s(nT)] \) or its inverse, we propose to use a joint estimator that is composed, at least conceptually, of an adaptive delay element \( D_a \) and a conventional \( M \)-order adaptive FIR filter with weight vector \( w_M(n) \). 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.

![Diagram](image)

**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) \( \xi_e = E[|e(n)|^2] \) with the help of gradient-based techniques (steepest-descent and LMS).

A simple adaptive filter has the potential to model the functional \( L_{D_a \ast s(nT)}[s(nT)] \) 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 \( D_a \) is modelled by a shift in the adaptive filter impulse response. For a fixed filter order \( M \), this shift may result in 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 \( \ell \), and the adaptive filter weight vector \( \mathbf{w}_M(n) \), of the sum of error squares

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

where \( M \) is the filter order and \( \lambda \) 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 \( \ell \). Then, the error \( e_M(i, d_i) \) in equation (4) can be expressed as

\[
e_M(i, d_i) = e_M(i, \ell) = r(i + \ell) - \mathbf{w}_M(n)^T \mathbf{u}_M(i), \tag{5}
\]

where \( H \) denotes complex conjugate transpose, \( r(n) \) is the output of the reference branch, \( \mathbf{u}(n) \) 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 \( \ell \), the minimum sum of weighted error squares \( \xi_M(n, \ell) \) is defined as

\[
\xi_M(n, \ell) = \min_{\mathbf{w}_M(n)} E_M(n). \tag{6}
\]

The weight vector for which this minimum is attained is defined as \( \mathbf{w}_M^*(n) \). If the adaptive delay \( \ell \) is not equal to \( D \), for all \( i \), the sum of errors \( \xi_M(n, d_i) \) is not minimum with respect to \( d_i \), 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 \( \mathbf{w}_M(n) \) and \( \xi_M(n, \ell) \).
2. Adapt \( \ell \) by using derivative information from \( \xi_M(n, \ell) \).

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 \( \ell \), of \( \xi_M(n, \ell) \). The gradient is given by

\[
\frac{\partial \xi_M(n, \ell)}{\partial \ell} = \begin{cases} 
1 & \text{if } \xi_M(n, \ell + 1) < \xi_M(n, \ell) \text{ and } \xi_M(n, \ell - 1) < \xi_M(n, \ell - 1) \\
0 & \text{otherwise,}
\end{cases} \tag{7}
\]

and the lag value is updated as

\[
\ell = \ell + \left( \frac{\partial \xi_M(n, \ell)}{\partial \ell} \right)^{+}, \tag{8}
\]

where \( \cdot \)^{+} denotes a time average. Note that when \( \ell \) is updated, \( \mathbf{w}_M(n) \) 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 \( \mathbf{w}_M(n) \). This also implies some problems when the value of \( \ell \) 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 \( \xi_M(n, \ell + 1) \), \( \xi_M(n, \ell - 1) \), \( \mathbf{w}^{\ell+1}_M(n) \) and \( \mathbf{w}^{\ell-1}_M(n) \) from the knowledge of \( \mathbf{w}^\ell_M(n) \) and \( \mathbf{w}^\ell_M(n) \). These lag-recursive relationships involve some variables that are defined in the following list:

- \( a_{M-1}(n) \) and \( b_{M-1}(n) \) 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 squares, \( f_{M-1}(n) \) and \( b_{M-1}(n) \)
- \( v^{\ell+1}_{M-1}(n) \) and \( v^{\ell-1}_{M-1}(n) \) 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 \( \ell \), i.e.

\[
v^{\ell+1}_{M-1}(n) = \sum_{i=1}^{n} \lambda^{i-1} f_{M-1}(i) r^*(i + \ell) \tag{9}
\]

\[
v^{\ell-1}_{M-1}(n) = \sum_{i=1}^{n} \lambda^{i-1} b_{M-1}(i) r^*(i + \ell) \tag{10}
\]

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

\[
a_{M-1}(n, \ell) = r(n + \ell) - \mathbf{w}^\ell_M(n - 1) B_{M-1}(n) \tag{11}
\]

\( g_{M-1}(n) \) is the Kalman gain of order \( M - 1 \) appearing in the RLS algorithm

\[
|\mathbf{w}^\ell_M(n)\|_{M-1} \text{ is defined as the } (M-1)-\text{vector corresponding to the first components of } \mathbf{w}^\ell_M(n) \tag{12}
\]

\( |\mathbf{w}^\ell_M(n)\|_{M-1} \) is the \( (M-1)-\text{vector corresponding to the last components of } \mathbf{w}^\ell_M(n) \)

\( w_{M1}(n) \) and \( w_{M2}(n) \) are respectively the first and last component of \( \mathbf{w}^\ell_M(n) \)

The lag-recursive relationships are:

\[
\dot{\mathbf{w}}^\ell_M(n) = \dot{\mathbf{w}}^{\ell+1}_M(n) - \dot{\mathbf{w}}^{\ell-1}_M(n) \tag{13}
\]

\[
\dot{\xi}_M(n, \ell + 1) = \dot{\xi}_M(n, \ell) + \frac{|v^{\ell+1}_{M-1}(n)|^2}{F_{M-1}(n)} \tag{14}
\]

\[
\dot{\xi}_M(n, \ell - 1) = \dot{\xi}_M(n, \ell) + \frac{|v^{\ell-1}_{M-1}(n)|^2}{B_{M-1}(n)} \tag{15}
\]

\[
\dot{\xi}_M(n, \ell) = \dot{\xi}_M(n, \ell - 1) - \lambda^{-1} g_{M-1}(n, \ell) e_{M-1}(n, \ell) \tag{16}
\]

\[
\hat{\mathbf{w}}^\ell_M(n) = \hat{\mathbf{w}}^{\ell+1}_M(n) - \hat{\mathbf{w}}^{\ell-1}_M(n) \tag{17}
\]

\[
\mathbf{w}^{\ell+1}_M(n) = \mathbf{w}^\ell_M(n - 1) + a_{M-1}(n) \mathbf{w}^\ell_M(n) \tag{18}
\]

\[
\mathbf{w}^{\ell-1}_M(n) = \mathbf{w}^\ell_M(n - 1) - b_{M-1}(n) \mathbf{w}^\ell_M(n - 1, \ell + 1) \tag{19}
\]
The ones
is be
ht
the lag update
variables.

These successive applications of the forward lag recursions pro-

\[ \begin{align*}
\hat{w}_{M-1}(n) &= \left[ \begin{array}{c}
\hat{w}_{M-1}(n) \\
\hat{w}_{M-1}(n-1)
\end{array} \right] + \frac{\psi_{M-1}(n)}{F_{M-1}(n)} \left[ \begin{array}{c}
1 \\
-a_{M-1}(n)
\end{array} \right],
\end{align*} \tag{12} \]

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 \( v_{M-1}(n) \) and \( v_{M-1}(n) \), are intermediate variables available in
the implementation of the fast transversal filter (FTF) RLS
algorithm [4]. As for \( v_{M-1}(n) \) and \( v_{M-1}(n) \), 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 prelimi-

\[ \begin{align*}
\gamma_m(n) &= 1 + \lambda^{-1} g_{M}^{(1)}(n) u_{M}^{(n)}.
\end{align*} \]

The fourth filter is for the computation of \( \xi_m(n, \ell - 1) \) and
\( \psi_{M-1}(n-1) \). Notice that \( \xi_m(n, \ell - 1) \) is also obtained from
that filter, through the first relation of (9). A fifth filter, with
weight vector \( \hat{w}_{M-1}(n) \), is used to obtain \( \psi_{M-1}(n) \), from
which \( \xi_m(n, \ell) \) and \( \xi_{M-1}(n, \ell + 1) \) are computed. Finally a

\[ \begin{align*}
\xi_{M-1}(n, \ell + 1) &= \left[ \begin{array}{c}
\xi_{M}(n, \ell + 1) \\
\xi_{M}(n, \ell + 1)
\end{array} \right],
\end{align*} \]

Fig. 2 Interpretation of the computation of the sums
of squared errors in terms of transversal filters
sixth transversal filter, with weight vector \( \hat{w}_{M-1}(n-1) \), is used in
the computation of \( \psi_{M-1}(n) \) and \( \xi_m(n, \ell - 1) \).

The originality of this joint LS algorithm resides in the
serial computations, from \( \psi_{M-1}(n) \), of all the necessary errors
and weight vectors for lags \( \ell \) 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 \( \hat{w}_{M-1}^{(n)}(n) \) or \( \hat{w}_{M-1}^{(n-1)}(n) \) 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 imple-
menced, one for each possible lag, in order to compute
\( \xi_m(n, \ell - 1) \), \( \xi_m(n, \ell) \), and \( \xi_m(n, \ell + 1) \). But this form of joint
algorithm does not allow, when the lag is updated at time \( n \),
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 er-
ror in both quantities, because their computation involves the
internal variables \( \gamma_m(n) \) and \( g_m(n) \), that were obtained from
nonzero initial conditions at \( n \). 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 filter-
ing 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 peri-
dic restart of the algorithm, similar to the technique proposed
by Eleftherious 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 imple-
menced by accumulating, for \( \ell, \ell + 1 \) and \( \ell - 1 \), the sums of
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 \( D_r \) 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. 4** Variations of the sum of squared errors with \( \ell \); \( \lambda = 0.97 \); (a) \( \xi_M(n, \ell - 1) \); (b) \( \xi_M(n, \ell) \); (c) \( \xi_M(n, \ell + 1) \) 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 0.7\( \pi \). The transmitted signal \( s(n) \) was a zero-mean process with a white power spectral density from \(-\pi\) to \(\pi\). 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 \( \ell \) (curve (b)) and \( \ell - 1 \) (curve (a)), indicating that the adaptive filter modelling becomes less effective for these delays.

**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**