Fast Algorithms with low Complexity for Adaptive Filtering

Madjid AREZKI[1], Daoud BERKANI[2]
[1]LATSI Laboratory – Department of Electronic
University of BLIDA, ALGERIA
md_arezki@hotmail.com

[2]Signal &Communications Laboratory – Department of Electrical Engineering
ENP, ALGIERS, ALGERIA.
dberkani@hotmail.com

Abstract: — The numerically stable version of fast recursive least squares (NS-FRLS) algorithms represent a very important load of calculation that needs to be reduced. Its computational complexity is of 8*L* operations per sample, where *L* is the finite impulse response filter length. We propose an algorithm for adaptive filtering, while maintaining equilibrium between its reduced computational complexity and its adaptive performances. We present a new (M-SMFTF) algorithm for adaptive filtering with fast convergence and low complexity. It is the result of a simplified FTF type algorithm, where the adaptation gain is obtained only from the forward prediction variables and using a new recursive method to compute the likelihood variable. This algorithm presents a certain interest, for the adaptation of very long filters, like those used in the problems of echo acoustic cancellation, due to its reduced complexity, its numerical stability and its convergence in the presence of the speech signal. Its computational complexity is of 6*L* and this is considerably reduced to 2*L*+4*P* when we use a reduced *P*-size (*P*<<*L*) forward predictor.

Key-Words: Adaptive Filters, FIR model, Fast Algorithms, Stability, Convergence Speed, Tracking capability.

1 Introduction

In general the problem of system identification involves constructing an estimate of an unknown system given only two signals, the input signal and a reference signal. Typically the unknown system is modelled linearly with a finite impulse response (FIR), and adaptive filtering algorithms are employed to iteratively converge upon an estimate of the response. If the system is time-varying, then the problem expands to include tracking the unknown system as it changes over time [1]-[3]. There are two major classes of adaptive algorithms. One is the least mean square (LMS) algorithm, which is based on a stochastic gradient method. The LMS algorithm has been extensively studied, and many theoretical results on its transient and steady state performances have been obtained [3]. The LMS algorithm has a computational complexity of $O(L)$, L is the FIR filter length. The other class of adaptive algorithm is the recursive least-squares (RLS) algorithm which minimizes a deterministic sum of squared errors [4]. The RLS algorithm solves this problem, but at the expense of increased computational complexity of $O(L^2)$. A large number of fast RLS (FRLS) algorithms have been developed over the years, but, unfortunately, it seems that the better a FRLS algorithm is in terms of computational efficiency, the more severe is its problems related to numerical stability [4]. Fast versions of these algorithms, namely, the fast Kalman [5], the fast a posteriori error sequential technique (FAEST) [6], and fast transversal filter (FTF) [7] algorithms, are derived from the RLS by the introduction of forward and backward predictors. The FRLS algorithm shows a complexity of $O(L)$. Several numerical solutions of stabilization, with stationary signals, are proposed in the literature [8]–[13]. Another way of reducing the complexity of the fast RLS (FRLS) algorithm has been proposed in [14], [15]: When the input signal can be accurately modelled by a predictor of order P, the fast Newton transversal filter (FNTF) avoids running forward and backward predictors of order L, which would be required by a FRLS algorithm. The required quantities are extrapolated from the predictors of order P (P<<L). Thus, the complexity of the FNTF falls down to (2L+12P) multiplications instead of 8L. Further complexity reduction in the prediction part calculation of the FNTF algorithm can be achieved by using fast backward prediction based algorithm (FPLS) [16]. The computational requirement of this latter algorithm is (2L+5P+1.5P^2). Recently, the simplified FTF-type algorithm [17] developed for use in acoustic echo cancellers. This algorithm derived from the FTF

algorithm where the adaptation gain is obtained only from the forward prediction variables. The computational complexity of this algorithm is 7*L* when used with a full size predictor which is less complex than the original numerically stable 8*L* FTF algorithm.

In this paper, we propose more complexity reduction of the simplified FTF-type algorithm by using a new recursive method to compute the likelihood variable. The computational complexity of the proposed algorithm is 6*L* and this computational complexity can be significantly reduced to (2*L*+4*P*) when used with a reduced *P*-size forward predictor. The M-SMFTF of the proposed algorithm outperforms the classical adaptive algorithms because of its convergence speed which approaches that of the RLS algorithm and its computational complexity which is slightly greater than the one of the NLMS algorithm. We describe the NLMS and numerically stable FRLS (NS-FRLS) algorithms. More complexity reduction for simplified FTF-type (M-SMFTF) algorithm is proposed. At the end, we present some simulation results of the M-SMFTF algorithm.

2 Adaptive Algorithms

The main identification block diagram of a linear system with finite impulse response (FIR) is represented in Fig.1.

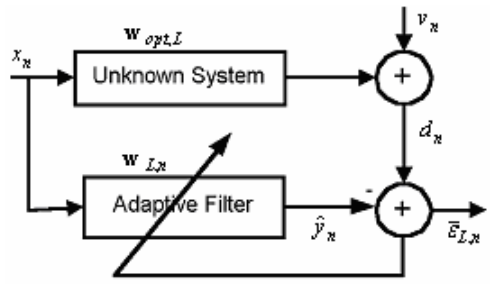

Fig.1: Main block diagram of an adaptive filter

The output a priori error $\bar{\varepsilon}_{L,n}$ of this system at time *n* is:

$$\bar{\varepsilon}_{L,n} = d_n - \hat{y}_n \tag{1}$$

where $\hat{y}_n = \mathbf{w}_{L,n-1}^{\mathrm{T}} \mathbf{x}_{L,n}$ is the model filter output, $\mathbf{x}_{L,n} = [x_n, x_{n-1}, \ldots, x_{n-L+1}]^{\mathrm{T}}$ is a vector containing the last L samples of the input signal x_n, $\mathbf{w}_{L,n-1} = [w_{1,n-1}, w_{2,n-1}, \ldots, w_{L,n-1}]^{\mathrm{T}}$ is the coefficient vector of the adaptive filter and L is the filter length. The desired signal from the model is:

$$d_n = v_n + \mathbf{w}_{opt,L}^{\mathrm{T}} \mathbf{x}_{L,n} \tag{2}$$

where $\mathbf{w}_{opt,L} = [w_{opt,1}, w_{opt,2}, \ldots, w_{opt,L}]^{\mathrm{T}}$ represents the unknown system impulse response vector and v_n is a stationary, zero-mean, and independent noise sequence that is uncorrelated with any other signal. The superscript $^{\mathrm{T}}$ describes transposition. The filter is updated at each instant by feedback of the estimation error proportional to the adaptation gain, denoted as $\mathbf{g}_{L,n}$, and according to:

$$\mathbf{w}_{L,n} = \mathbf{w}_{L,n-1} + \mathbf{g}_{L,n}\, \bar{\varepsilon}_{L,n} \tag{3}$$

The different algorithms are distinguished by the gain calculation.

2.1 The NLMS Algorithm

Algorithms derived from the gradient [3], for which the optimization criterion corresponds to a minimization of the mean-square error. For the normalized LMS (NLMS) algorithm, the adaptation gain is given by:

$$\mathbf{g}_{L,n} = \frac{\mu}{L\,\pi_{x,n} + c_0} \mathbf{x}_{L,n} \tag{4}$$

where μ is referred to as the adaptation step and c_0 is a small positive constant used to avoid division by zero in absence of the input signal. The stability condition of this algorithm is $0 < \mu < 2$ and the fastest convergence is obtained for $\mu = 1$ [18]. The power $\pi_{x,n}$ of input signal can alternatively be estimated using the following recursive equation [19]:

$$\pi_{x,n} = (1-\gamma)\,\pi_{x,n-1} + \gamma\, x_n^2 \tag{5}$$

where γ is a forgetting factor ($\gamma \approx 1/L$). The computational complexity of the NLMS algorithm is 2*L* multiplications per sample.

2.2 The NS-FRLS Algorithm

The filter $\mathbf{w}_{L,n}$ is calculated by minimizing the weighted least squares criterion according to [1]:

$$J_n(\mathbf{w}) = \sum_{i=1}^{n} \lambda^{n-i} \left(d_i - \mathbf{w}_{L,n}^T \mathbf{x}_{L,i}\right)^2 \tag{6}$$

where λ denotes the exponential forgetting factor ($0 < \lambda \leq 1$). The adaptation gain is given by:

$$\mathbf{g}_{L,n} = \begin{cases} \mathbf{R}_{L,n}^{-1} \mathbf{x}_{L,n}; & \text{RLS} \\ \gamma_{L,n} \widetilde{\mathbf{k}}_{L,n}; & \text{FRLS} \end{cases} \tag{7}$$

$$\mathbf{R}_{L,n} = \sum_{i=1}^{n} \lambda^{n-i} \mathbf{x}_{L,i} \mathbf{x}_{L,i}^T = \lambda\, \mathbf{R}_{L,n-1} + \mathbf{x}_{L,n} \mathbf{x}_{L,n}^{\mathrm{T}} \tag{8}$$

where $\mathbf{R}_{L,n}$ is an estimate of the correlation

matrix of the input signal vector. The variables $\gamma_{L,n}$ and $\widetilde{\mathbf{k}}_{L,n}$ respectively indicate the likelihood variable and normalized Kalman gain vector. This latter is calculated, independently of the filtering part $\mathbf{w}_{L,n}$, by a FRLS algorithm using forward/backward linear prediction analysis over the signal x_n [1]. The calculation complexity of a FRLS algorithm is of order L. This reduction of complexity, compared to that of RLS algorithms, which have a complexity of order L^2, have made all FRLS algorithms numerically unstable.

The numerical stability is achieved by using a control variable, called also a divergence indicator ξ_n [11], theoretically equals to zero. Its introduction in an unspecified point of the algorithm modifies its numerical properties. It is obtained by using some redundant formulae of the FRLS algorithms.This variable is given by:

$$\xi_n = \bar{r}_{L,n} - \bar{r}^{\,\mathrm{f}}_{L,n} \begin{cases} = 0 & \text{theory} \\ \neq 0 & \text{practical} \end{cases} \tag{9}$$

$$\bar{r}^{\,\mathrm{f}}_{L,n} = [(1-\mu_s)\bar{r}^{\,\mathrm{f}_0}_{L,n} + \mu_s \bar{r}^{\,\mathrm{f}_1}_{L,n}] \tag{10a}$$

$$0 \le \mu_s \le 1 \tag{10b}$$

where ($\bar{r}_{L,n}$, $\bar{r}^{\,\mathrm{f}_0}_{L,n}$ and $\bar{r}^{\,\mathrm{f}_1}_{L,n}$) are the backward a priori prediction errors calculate differently in tree ways. We define three backward a priori prediction errors, theoretically equivalents, which will be used to calculate the likelihood variable $\gamma_{L,n}$, the backward prediction error variance $\beta_{L,n}$ and the backward prediction $\mathbf{b}_{L,n}$. We introduce these variables into the algorithm, and we use suitably the scalar parameters $(\mu^{\gamma}, \mu^{\beta}, \mu^{b})$ and μ_s, in order to obtain the numerical stability. For appropriate choices, we selected the following control parameters:

$$\mu^{\gamma} = 0,\ \mu^{\beta} = \mu^{b} = 1;\ \mu_s = 0.5 \tag{11}$$

It can be shown that the variance of the numerical errors in the backward predictor, with the assumption of a white Gaussian input signal, is stable under the following condition [11]:

$$\lambda > \frac{4L+5}{4L+7} = 1 - \frac{1}{2L+3.5} \tag{12a}$$

These conditions can be written in another simpler form:

$$\lambda = 1 - 1/2L \tag{12b}$$

The resulting stabilized FRLS (NS-FRLS) algorithms have a complexity of 8L; it is given in Table 1.

Table1: NS-FRLS (8L) algorithm

Initialization: : $E_0 \ge \sigma_x^2 L/100$; $\gamma_{L,0} = 1$; $\alpha_{L,0} = \lambda^L E_0$; $\beta_{L,0} = E_0$;

$\mathbf{w}_{L,0} = \mathbf{a}_{L,0} = \mathbf{b}_{L,0} = \widetilde{\mathbf{k}}_{L,0} = 0_L$.

Variables available at the discrete-time index n:

$\mathbf{a}_{L,n-1}; \mathbf{b}_{L,n-1}; \widetilde{\mathbf{k}}_{L,n-1}; \gamma_{L,n-1}; \alpha_{L,n-1}; \beta_{L,n-1}; \mathbf{w}_{L,n-1}$

New information: x_n , d_n .

- Prediction Part:

Modeling of x_n , x_{n-L}

$\bar{e}_{L,n} = x_n - \mathbf{a}^{\mathrm{T}}_{L,n-1}\mathbf{x}_{L,n-1}$;

$$\widetilde{\mathbf{k}}^{+}_{L+1,n} = \begin{bmatrix} \widetilde{\mathbf{k}}^{+}_{L,n} \\ \widetilde{k}^{+}_{L+1,n} \end{bmatrix} = \begin{bmatrix} 0 \\ \widetilde{\mathbf{k}}_{L,n-1} \end{bmatrix} + \frac{\bar{e}_{L,n}}{\lambda \alpha_{L,n-1}} \begin{bmatrix} 1 \\ -\mathbf{a}_{L,n-1} \end{bmatrix};$$

$\mathbf{a}_{L,n} = \mathbf{a}_{L,n-1} + \bar{e}_{L,n}\gamma_{L,n-1}\widetilde{\mathbf{k}}_{L,n-1}$; $\alpha_{L,n} = \lambda\alpha_{L,n-1} + \gamma_{L,n-1}\bar{e}^2_{L,n}$

$\bar{r}_{L,n} = x_{n-L} - \mathbf{b}^{\mathrm{T}}_{L,n-1}\mathbf{x}_{L,n}$; $\bar{r}^{\,\mathrm{f}_0}_{L,n} = \lambda\beta_{L,n-1}\widetilde{k}^{+}_{L+1,n}$; $\bar{r}^{\,\mathrm{f}_1}_{L,n} = \lambda^{-L+1}\gamma_{L,n-1}\alpha_{L,n-1}\widetilde{k}^{+}_{L+1,n}$

$\xi_n = \bar{r}_{L,n} - [(1-\mu_s)\bar{r}^{\,\mathrm{f}_0}_{L,n} + \mu_s\bar{r}^{\,\mathrm{f}_1}_{L,n}]$;

$\bar{r}^{\,\gamma}_{L,n} = \bar{r}_{L,n} + \mu^{\gamma}\ \xi_n$; $\bar{r}^{\,\beta}_{L,n} = \bar{r}_{L,n} + \mu^{\beta}\ \xi_n$; $\bar{r}^{\,b}_{L,n} = \bar{r}_{L,n} + \mu^{b}\ \xi_n$;

$\widetilde{\mathbf{k}}_{L,n} = \widetilde{\mathbf{k}}^{+}_{L,n} + \widetilde{k}^{+}_{L+1,n}\mathbf{b}_{L,n-1}$; $\gamma_{L,n} = \dfrac{\lambda\alpha_{L,n-1}}{\alpha_{L,n} - \lambda^L(\bar{r}^{\,\gamma}_{L,n})^2}\gamma_{L,n-1}$;

$\mathbf{b}_{L,n} = \mathbf{b}_{L,n-1} + \bar{r}^{\,b}_{L,n}\gamma_{L,n}\widetilde{\mathbf{k}}_{L,n}$; $\beta_{L,n} = \lambda\beta_{L,n-1} + \gamma_{L,n}(\bar{r}^{\,\beta}_{L,n})^2$;

- Filtering Part:

$\bar{\varepsilon}_{L,n} = d_n - \mathbf{w}^{\mathrm{T}}_{L,n-1}\mathbf{x}_{L,n}$; $\mathbf{w}_{L,n} = \mathbf{w}_{L,n-1} + \bar{\varepsilon}_{L,n}\gamma_{L,n}\widetilde{\mathbf{k}}_{L,n}$

Note that numerical stabilization of the algorithm limits the range of the forgetting factor λ (condition (12)) and consequently their convergence speed and tracking ability.

2.3 The M-SMFTF Algorithm

We propose a new (M-SMFTF) algorithm for adaptive filtering with fast convergence and low complexity. We present more complexity reduction of the simplified FTF-type algorithm by using a new recursive method to compute the likelihood variable. The simplified FTF-type algorithm [17] derived from the FTF algorithm where the adaptation gain is obtained only from the forward prediction variables and the likelihood variable is given by using the definition directly:

$$\gamma_{L,n} = \frac{1}{1 + \widetilde{\mathbf{k}}^{\mathrm{T}}_{L,n}\mathbf{x}_{L,n}} \tag{13}$$

The backward prediction variables, which are the main source of the numerical instability in the FRLS algorithms [8], [9], [11] and [12], are completely discarded. By using only forward prediction variables and adding a small regularization constant c_a and a leakage factor η, we obtain a robust numerically stable adaptive algorithm that shows the same performances as FRLS algorithms.

By taking the expression of normalized Kalman gain:

$$\begin{bmatrix}\widetilde{\mathbf{k}}_{L,n}\\ 0\end{bmatrix}=\begin{bmatrix}0\\ \widetilde{\mathbf{k}}_{L,n-1}\end{bmatrix}+\frac{\bar{e}_{L,n}}{\lambda\alpha_{L,n-1}}\begin{bmatrix}1\\ -\mathbf{a}_{L,n-1}\end{bmatrix}-\frac{\bar{r}_{L,n}}{\lambda\beta_{L,n-1}}\begin{bmatrix}-\mathbf{b}_{L,n-1}\\ 1\end{bmatrix} \quad (14)$$

and if we discard all backward prediction variables from (14) and use only the forward variables to compute the normalized Kalman gain:

$$\begin{bmatrix}\widetilde{\mathbf{k}}_{L,n}\\ *\end{bmatrix}=\begin{bmatrix}0\\ \widetilde{\mathbf{k}}_{L,n-1}\end{bmatrix}+\frac{\bar{e}_{L,n}}{\lambda\alpha_{L,n-1}}\begin{bmatrix}1\\ -\mathbf{a}_{L,n-1}\end{bmatrix} \quad (15a)$$

$$\mathbf{a}_{L,n}=\mathbf{a}_{L,n-1}+\bar{e}_{L,n}\gamma_{L,n-1}\widetilde{\mathbf{k}}_{L,n-1} \quad (15b)$$

This algorithm is not very robust with nonstationarity input signal like speech signals. The first difficulty comes from $\alpha_{L,n}=\lambda\alpha_{L,n-1}\to 0$. This convergence to zero puts FTF algorithms and their numerically stable versions in very difficult situations. Instability may occur since we are trying to perform numerical divisions by very small values. To guard against this possibility, like it is often done with the NLMS algorithm, we append a small positive constant c_a to the denominator

$$\frac{\bar{e}_{L,n}}{\lambda\alpha_{L,n-1}}\to\frac{\bar{e}_{L,n}}{\lambda\alpha_{L,n-1}+c_a} \quad (16)$$

The second difficulty is that the forward predictor is locked over its last values. It is known that the FRLS algorithms were developed in the prewindowing case and all vectors are initialised by zero so that the algorithm starts adapting. In these conditions, when the input signal vanishes and reappears after a long period of time, the algorithm may diverge because of these nonzero values of the predictor. In other words, the algorithm is not well initialised when the signal reappears. In such conditions, it might be preferable to have the forward predictor $\mathbf{a}_{L,n}$ return back to zero by doing the following operation:

$$\mathbf{a}_{L,n}\to\eta\,\mathbf{a}_{L,n} \quad (17)$$

where η is a close to one constant often called the leakage factor [13].

Let us replace the quantity (*), that has not been used in $\widetilde{\mathbf{k}}_{L,n}$ of (15a), by the variable $c_{L,n}$, we obtain:

$$\begin{bmatrix}\widetilde{\mathbf{k}}_{L,n}\\ c_{L,n}\end{bmatrix}=\begin{bmatrix}0\\ \widetilde{\mathbf{k}}_{L,n-1}\end{bmatrix}+\frac{\bar{e}_{L,n}}{\lambda\alpha_{L,n-1}+c_a}\begin{bmatrix}1\\ -\mathbf{a}_{L,n-1}\end{bmatrix} \quad (18)$$

By exploiting certain invariance properties by shifting the vector input signal extended to the order (L+1), we obtain two writing manners of input vector:

$$\mathbf{x}_{L+1,n}=\left[\mathbf{x}_{L,n}^{\mathrm{T}},x_{n-L}\right]^{\mathrm{T}} \quad (19a);\quad \mathbf{x}_{L+1,n}=\left[x_n,\mathbf{x}_{L,n-1}^{\mathrm{T}}\right]^{\mathrm{T}} \quad (19b)$$

By multiplying on the left, the members of left and right of the expression (18) by equations (19a) and (19b) respectively, the following equality is obtained:

$$\mathbf{x}_{L,n}^{\mathrm{T}}\widetilde{\mathbf{k}}_{L,n}+c_{L,n}\,x_{n-L}=\mathbf{x}_{L,n-1}^{\mathrm{T}}\widetilde{\mathbf{k}}_{L,n-1}+\frac{\bar{e}_{L,n}^2}{\lambda\alpha_{L,n-1}+c_a} \quad (20)$$

By manipulating the relation (20), we obtain a new recursive formula for calculating the likelihood variable as given below:

$$\gamma_{L,n}=\frac{\gamma_{L,n-1}}{1+\delta_{L,n}\gamma_{L,n-1}} \quad (21)$$

where

$$\delta_{L,n}=\frac{\bar{e}_{L,n}^2}{\lambda\alpha_{L,n-1}+c_a}-c_{L,n}\,x_{n-L} \quad (22)$$

The computational complexity of the M-SMFTF algorithm is 6L, it is given in Table2.

Table2: M-SMFTF (6L) algorithm

Initialization::

$\mathbf{w}_{L,0}=\mathbf{a}_{L,0}=\widetilde{\mathbf{k}}_{L,0}=0_L$; $\gamma_{L,0}=1$; $\alpha_{L,0}=\lambda^L E_0$; ; $E_0\geq\sigma_x^2 L/100$

Variables available at the discrete-time index n:

$\mathbf{a}_{L,n-1};\widetilde{\mathbf{k}}_{L,n-1};\gamma_{L,n-1};\alpha_{L,n-1};\mathbf{w}_{L,n-1}$

New information: x_n , d_n .

- **Prediction Part:**

$\bar{e}_{L,n}=x_n-\mathbf{a}_{L,n-1}^{\mathrm{T}}\mathbf{x}_{L,n-1}$;

$\begin{bmatrix}\widetilde{\mathbf{k}}_{L,n}\\ c_{L,n}\end{bmatrix}=\begin{bmatrix}0\\ \widetilde{\mathbf{k}}_{L,n-1}\end{bmatrix}+\frac{\bar{e}_{L,n}}{\lambda\alpha_{L,n-1}+c_a}\begin{bmatrix}1\\ -\mathbf{a}_{L,n-1}\end{bmatrix}$;

$\mathbf{a}_{L,n}=\eta\left\{\mathbf{a}_{L,n-1}+\bar{e}_{L,n}\gamma_{L,n-1}\widetilde{\mathbf{k}}_{L,n-1}\right\}$; $\alpha_{L,n}=\lambda\alpha_{L,n-1}+\gamma_{L,n-1}\bar{e}_{L,n}^2$

$\delta_{L,n}=\frac{\bar{e}_{L,n}^2}{\lambda\alpha_{L,n-1}+c_a}-c_{L,n}\,x_{n-L}$; $\gamma_{L,n}=\frac{\gamma_{L,n-1}}{1+\delta_{L,n}\gamma_{L,n-1}}$

- **Filtering Part:**

$\bar{\varepsilon}_{L,n}=d_n-\mathbf{w}_{L,n-1}^{\mathrm{T}}\mathbf{x}_{L,n}$; $\mathbf{w}_{L,n}=\mathbf{w}_{L,n-1}+\bar{\varepsilon}_{L,n}\gamma_{L,n}\widetilde{\mathbf{k}}_{L,n}$

2.4 The RM-SMFTF Algorithm

The Reduced size predictors in the FTF algorithms have been successfully used in the FNTF algorithms [14], [15]. By a method of extrapolation, the autocorrelation matrix of order L is built starting from an estimate of the autocorrelation matrix of order P (P<<L). The proposed algorithm can be easily used with reduced size prediction part. If we denote P the order of the predictor and L the size of adaptive filter, the forward predictor and the normalized Kalman gain are given respectively by:

$$\mathbf{a}_{L,n} = \begin{bmatrix} \mathbf{a}_{P,n} \\ 0_{L-P} \end{bmatrix} \quad (23a)$$

$$\begin{bmatrix} \widetilde{\mathbf{k}}_{L,n} \\ c_{L,n} \end{bmatrix} = \begin{bmatrix} 0 \\ \widetilde{\mathbf{k}}_{L,n-1} \end{bmatrix} + \frac{\bar{e}_{P,n}}{\lambda\alpha_{P,n-1} + c_a} \begin{bmatrix} 1 \\ -\mathbf{a}_{P,n-1} \\ 0_{L-P} \end{bmatrix} \quad (23b)$$

where P is much smaller than L. The first (P+1) components of the $\widetilde{\mathbf{k}}_{L,n}$ are updated using the reduced size forward variables, the last components are just a shifted version of the $(P+1)^{\text{th}}$ component of $\widetilde{\mathbf{k}}_{L,n}$. For this algorithm, we need two likelihood variables: the first one $\gamma_{P,n}$, is used to update the forward prediction error variance $\alpha_{P,n}$:

$$\gamma_{P,n} = \frac{\gamma_{P,n-1}}{1 + \delta_{P,n}\gamma_{P,n-1}} \quad (24a)$$

$$\delta_{P,n} = \frac{\bar{e}_{P,n}^2}{\lambda\alpha_{P,n-1} + c_a} - c_{P,n}\, x_{n-P} \quad (24b)$$

where $c_{P,n}$ is $(P+1)^{\text{th}}$ component of $\widetilde{\mathbf{k}}_{L,n}$. The second likelihood variable $\gamma_{L,n}$, is used to update the forward predictor $\mathbf{a}_{P,n}$ of order P and the transversal filter $\mathbf{w}_{L,n}$:

$$\gamma_{L,n} = \frac{\gamma_{L,n-1}}{1 + \delta_{L,n}\gamma_{L,n-1}} \quad (25a)$$

$$\delta_{L,n} = \frac{\bar{e}_{P,n}^2}{\lambda\alpha_{P,n-1} + c_a} - c_{L,n}\, x_{n-L} \quad (25b)$$

The computational complexity of this algorithm is ($2L$+$4P$); it is given in Table 3.

Table3: RM-SMFTF ($2L$+$4P$) algorithm

Initialization:: $E_0 \geq \sigma_x^2 P/100$;

$\gamma_{P,0} = 1$; $\alpha_{P,0} = \lambda^P E_0$; $\gamma_{L,0} = 1$; $\mathbf{w}_{L,0} = \widetilde{\mathbf{k}}_{L,0} = 0_L$; $\mathbf{a}_{P,0} = 0_P$.

Variables available at the discrete-time index n:

$\mathbf{a}_{L,n-1}; \widetilde{\mathbf{k}}_{L,n-1}; \gamma_{L,n-1}; \alpha_{L,n-1}; \mathbf{w}_{L,n-1}$;

New information: x_n , d_n .

- Prediction Part:

$\bar{e}_{P,n} = x_n - \mathbf{a}_{P,n-1}^{\mathrm{T}} \mathbf{x}_{P,n-1}$;

$$\begin{bmatrix} \widetilde{\mathbf{k}}_{L,n} \\ c_{L,n} \end{bmatrix} = \begin{bmatrix} 0 \\ \widetilde{\mathbf{k}}_{L,n-1} \end{bmatrix} + \frac{\bar{e}_{P,n}}{\lambda\alpha_{P,n-1} + c_a} \begin{bmatrix} -1 \\ \mathbf{a}_{P,n-1} \\ 0_{L-P} \end{bmatrix};$$

$\widetilde{\mathbf{k}}_{P,n-1} = \widetilde{\mathbf{k}}_{L,n-1}(1:P)$; $c_{P,n} = \widetilde{\mathbf{k}}_{L,n}(P+1)$;

$\mathbf{a}_{P,n} = \eta\left\{\mathbf{a}_{P,n-1} + \bar{e}_{P,n}\gamma_{L,n-1}\widetilde{\mathbf{k}}_{P,n-1}\right\}$; $\alpha_{P,n} = \lambda\alpha_{P,n-1} + \gamma_{P,n-1}\bar{e}_{P,n}^2$

$\delta_{P,n} = \dfrac{\bar{e}_{P,n}^2}{\lambda\alpha_{P,n-1} + c_a} - c_{P,n}\, x_{n-P}$; $\gamma_{P,n} = \dfrac{\gamma_{P,n-1}}{1 + \delta_{P,n}\gamma_{P,n-1}}$;

$\delta_{L,n} = \dfrac{\bar{e}_{P,n}^2}{\lambda\alpha_{P,n-1} + c_a} - c_{L,n}\, x_{n-L}$; $\gamma_{L,n} = \dfrac{\gamma_{L,n-1}}{1 + \delta_{L,n}\gamma_{L,n-1}}$;

- Filtering Part:

$\bar{\varepsilon}_{L,n} = d_n - \mathbf{w}_{L,n-1}^{\mathrm{T}} \mathbf{x}_{L,n}$; $\mathbf{w}_{L,n} = \mathbf{w}_{L,n-1} + \bar{\varepsilon}_{L,n}\gamma_{L,n}\widetilde{\mathbf{k}}_{L,n}$

2.5 Analysis Prediction Part

We study the errors propagation in all recursive quantities of the prediction part of the M-SMFTF algorithm. Assuming that the numerical errors are small, the error propagation model in the recursive variables can be approximated by the following linear model [9]:

$$\Delta\boldsymbol{\varphi}_n = \mathbf{F}(n)\,\Delta\boldsymbol{\varphi}_{n-1} \quad (26)$$

where

$$\Delta\boldsymbol{\varphi}_n = \left[(\Delta\mathbf{a}_{L,n}^{\mathrm{T}}, \Delta\alpha_{L,n}),\ (\Delta\widetilde{\mathbf{k}}_{L,n}^{\mathrm{T}}, \Delta\gamma_{L,n})\right]^{\mathrm{T}} \quad (27)$$

represent the errors cumulated up until the time n in the forward and Kalman recursive variables, and the ($2L$+2)x($2L$+2) dimensional matrix $\mathbf{F}(n)$ given by:

$$\mathbf{F}(n) = \begin{bmatrix} \mathbf{F}_{11}(n) & \mathbf{F}_{12}(n) \\ \mathbf{F}_{21}(n) & \mathbf{F}_{22}(n) \end{bmatrix} \quad (28)$$

represents the transition matrix. The system (26) is said to be stable, in the mean sense, if the eigenvalues of $\mathrm{E}\{\mathbf{F}(n)\}$, in the steady state, are all less than one in magnitude [9]. The operator $\mathrm{E}\{.\}$ denotes the expected value.

We approximate the errors in the forward variables ($\Delta\mathbf{a}_{L,n}, \Delta\alpha_{L,n}$) and the Kalman variables ($\Delta\widetilde{\mathbf{k}}_{L,n-1}, \Delta\gamma_{L,n-1}$) by the following linear first order models deduced from differentiating ($\mathbf{a}_{L,n}, \alpha_{L,n}$) and ($\widetilde{\mathbf{k}}_{L,n}, \gamma_{L,n}$) respectively:

$$\Delta\mathbf{a}_{L,n} = \eta\left(\mathbf{I}_L - \gamma_{L,n-1}\widetilde{\mathbf{k}}_{L,n-1}\mathbf{x}_{L,n-1}^{\mathrm{T}}\right)\Delta\mathbf{a}_{L,n-1} + \mathbf{p}_a(n) \quad (29a)$$

$$\Delta\alpha_{L,n} = \lambda\,\Delta\alpha_{L,n-1} + p_\alpha(n) \quad (29b)$$

$$\Delta\widetilde{\mathbf{k}}_{L,n} = \mathbf{M}^k\,\Delta\widetilde{\mathbf{k}}_{L,n-1} + \mathbf{p}_k(n) \quad (29c)$$

$$\Delta\gamma_{L,n} = c^{\gamma}(n)\,\Delta\gamma_{L,n-1} + p_\gamma(n) \quad (29d)$$

where $\mathbf{p}_a(n)$, $p_\alpha(n)$, $\mathbf{p}_k(n)$ and $p_\gamma(n)$ represent the perturbation terms; and where

$$\mathbf{M}^k = \begin{bmatrix} 0_{L-1}^{\mathrm{T}} & 0 \\ \mathbf{I}_{L-1} & 0_{L-1} \end{bmatrix} \quad (30)$$

$$c^{\gamma}(n) = \frac{\gamma_{L,n}}{\gamma_{L,n-1}}(1 - \delta_{L,n}\gamma_{L,n}) \quad (31)$$

In asymptotic mode and by considering the averaging analysis, we can write:

$$\eta\left(\mathbf{I}_L - \gamma_{L,n-1}\widetilde{\mathbf{k}}_{L,n-1}\mathbf{x}_{L,n-1}^{\mathrm{T}}\right) \to \eta\,\lambda\,\mathbf{I}_L \quad (32)$$

$$\mathrm{E}\left\{c^{\gamma}(n)\right\} \to \frac{1}{\lambda^{-1} + \lambda - 1} \quad (33)$$

By assuming that, the perturbation terms remain limited. We can thus say that the system is

numerically stable, in the mean sense, for λ and η between zero and one.

We note that, the necessary condition of stability is the limit of the errors variance in the forward prediction. Let us calculate the covariance matrix of numerical errors in forward predictor. For that, we use the approach statistical $\mathbf{A}_n = \mathrm{E}\{\Delta\mathbf{a}_{L,n}\Delta\mathbf{a}_{L,n}^{\mathrm{T}}\}$. We assume that the components of vector $\Delta\mathbf{a}_{L,n}$ are independent between them and independent of the various theoretical variables given in the algorithm, and we suppose input signal Gaussian sequence, we obtain theses expressions:

$$\mathbf{A}_n = G\ \mathbf{A}_{n-1} + \mathrm{E}\{\mathbf{p}_a(n)\mathbf{p}_a^{\mathrm{T}}(n)\} \quad (34)$$

$$G = \eta^2\left(1-2(1-\lambda)+(1-\lambda)^2(L+2)\right) \quad (35)$$

The stability condition of equation (34) is given by the solution of the following inequality:

$$|G|<1 \quad (36)$$

This inequality is only verified for this condition:

$$\lambda > 1-\frac{1+\sqrt{1+\left(\frac{1}{\eta^2}-1\right)(L+2)}}{(L+2)} \quad (37)$$

We note that the lower bound of this condition is always smaller than the lower bound of condition (12) of the original numerically stable FRLS algorithm, which means that we can choose smaller values for the forgetting factor for the proposed algorithm and consequently have faster convergence rate and better tracking ability.

3 Simulation results

To confirm the validity of our analysis and demonstrate the improved numerical performance, some simulations are carried out. All plots show the mean squared modelling versus the number of iterations. For the purpose of smoothing the curves, error samples are averaged over 256 points. The forgetting factor λ and the leakage factor η for the M-SMFTF algorithm are chosen according to (37) with the stationary input. The regularisation constant c_a is used to limit the dynamic of the adaptation gain: large values give a more robust algorithm to the nonstationary of the input signal, small values of c_a allow more dynamic of the adaptation gain and may improve the convergence speed. In our experiments, we have used values of c_a comparable with the input signal power.

3.1 Comparative performances of algorithms

We define the norm gain-error variable $NGE(n)$ by:

$$NGE(n) = 10\log_{10}\left(\mathrm{E}\left\{\left\|\Delta\mathbf{g}_{L,n}\right\|^2\right\}\right) \quad (38)$$

where $\Delta\mathbf{g}_{L,n} = \left(\gamma_{L,n}^d\tilde{\mathbf{k}}_{L,n} - \gamma_{L,n}^f\tilde{\mathbf{k}}_{L,n}\right)$ is gain-error vector, $\gamma_{L,n}^d$ and $\gamma_{L,n}^f$ are likelihood variables calculated by the simplified FTF-type (7L) algorithm given in [17] and the proposed algorithm M-SMFTF (6L) respectively. We have simulated the algorithms to verify their correctness. The input signal x_n used in our simulation is a white Gaussian noise, with mean zero and variance one. The filter length is L=32, we run the algorithms with a forgetting factor λ=0.9688, the leakage factor η=0.98 and c_a=0.1. In Fig.2, we give the evolution in decibels of the norm gain-error $NGE(n)$; we can see that the round-off error signal stays constant. These algorithms produce exactly the same filtering error signal.

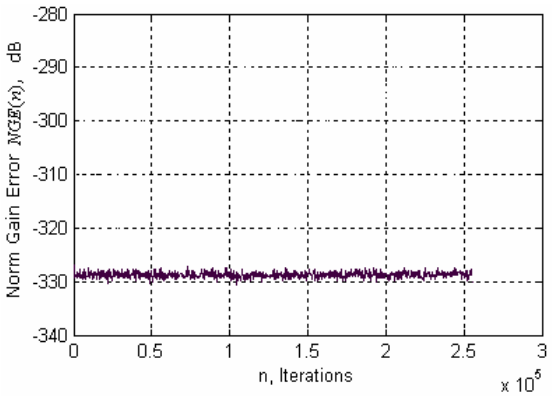

Fig.2: Evolution of the norm gain-error $NGE(n)$ for L=32, λ=0.9688, η=0.98, c_a=0.1, E_0=0.5

3.2 Comparative performances for stationary signals

We used a stationary correlated noise with a spectrum equivalent to the average spectrum of speech, called USASI noise in the field of acoustic echo cancellation. This signal, with mean zero and variance equal to 0.32, sampled at 16 kHz is filtered by impulse response which represents a real impulse response measured in a car and truncated to 256 samples. We compare the convergence speed and tracking capacity of the proposed algorithm with NS-FRLS and NLMS algorithms. The NLMS (μ=1) and NS-FRLS ($\lambda = 1-1/3L$) algorithms are tuned to obtain fastest convergence. The nonstationarity of the system to be modelled is

simulated by introducing a linear gain variation on the desired signal.

The filter length is L=256, the forgetting factor is ($\lambda = 1-1/L$) of the M-SMFTF algorithm. And for RM-SMFTF algorithm, the predictor order is P, the forgetting factor is ($\lambda = 1-1/P$).

Fig.3 shows that better performances in convergence speed are obtained for the M-SMFTF algorithm. The differences in the final $MSE(n)$ for the M-SMFTF and NS-FRLS algorithms are due to the use of different forgetting factors λ. It is observed that the proposed algorithm converges much faster and tracks better the variation of the system than both NS-FRLS and NLMS algorithms

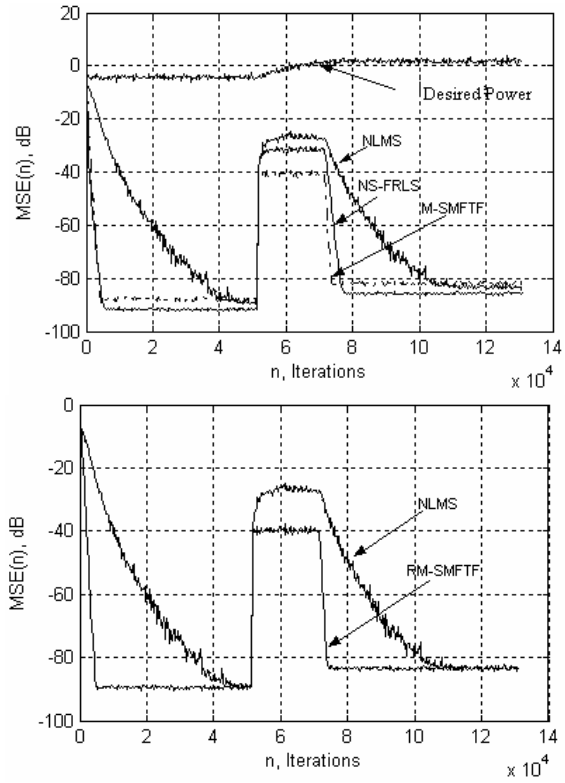

Fig.3: Comparative performance of the algorithms for USASI noise, L=256. M-SMFTF: λ=0.9961, η=0.985, c_a=0.5, E_0=1; NS-FRLS: λ=0.9987; NLMS: μ=1; RM-SMFTF:P=32, λ=0.9688, η=0.9985, c_a=0.5,E_0=0.2.

In Fig.4, we compare the convergence performance of the NLMS algorithm and RM-SMFTF algorithm with different values for the leakage η. We simulated an abrupt change in the impulse response at the 56320th samples. We use the following parameters: the predictor order is P=32, the forgetting factor is $\lambda = 1-1/P$.

The convergence speed of RM-SMFTF is much faster than NLMS. We notice, for the RM-SMFTF algorithm, that the more η approaches one and the better the speed.

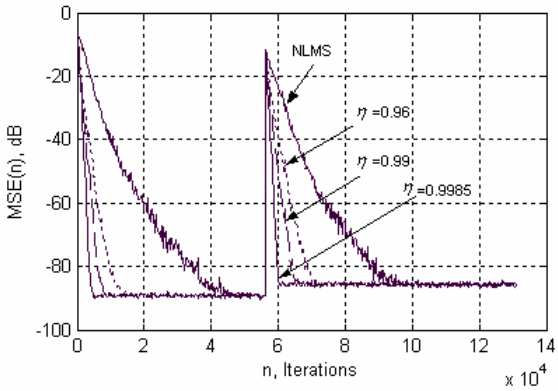

Fig.4: Comparative performance of the RM-SMFTF and NLMS for USASI noise, L=256, no output noise. RM-SMFTF: P=32, λ=0.9688, c_a=0.5, E_0=0.2, with different values for η; NLMS: μ=1.

3.3 Comparative performances for speech signals

The input signal used in the simulations is speech signal, sampled at 16 kHz. We compare the convergence speed of the proposed algorithm with NS-FRLS and NLMS algorithms. We simulated an abrupt change in the impulse response at the 56320th samples.

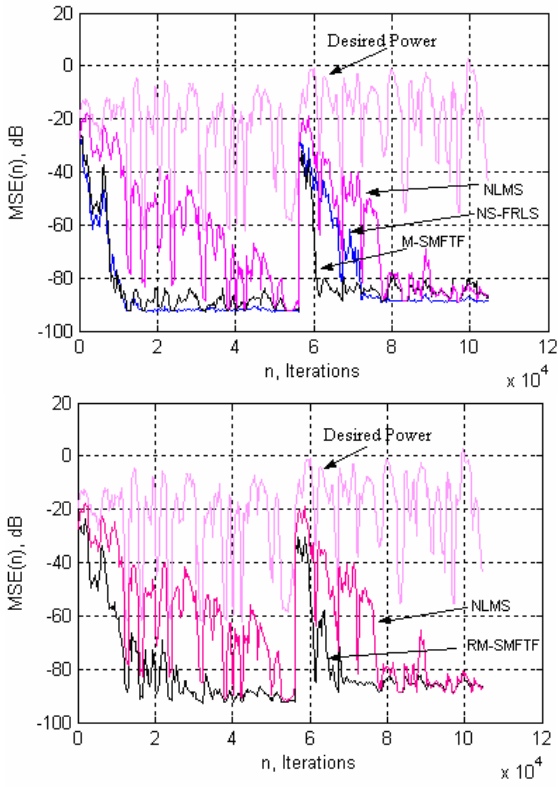

Fig.5: Comparative performance of the algorithms with speech input, L=256. M-SMFTF: λ=0.9961, η=0.96, c_a=0.1, E_0=0.5;NS-FRLS: λ=0.9996; NLMS: μ=1; RM-SMFTF: P=20, λ=0.950, η=0.99, c_a=0.1, E_0=1.

The choice of the forgetting factor for NS-FRLS algorithm to ensure numerical stability is $\lambda = 1 - 1/10L$. The forgetting factor for M-SMFTF algorithm is $\lambda = 1 - 1/L$, but the leakage η and the constant c_a must be carefully chosen. For the RM-SMFTF algorithm and the predictor order is *P*, the forgetting factor is ($\lambda = 1 - 1/P$).

In Fig.5, we can see that the initial convergence is almost the same for both M-SMFTF and NS-FRLS algorithms. But the M-SMFTF achieves better re-convergence after the abrupt change in the impulse response. From this plot, we observe that the re-convergence of RM-SMFTF is again faster than NLMS.

Different simulations have been done for different sizes *L* and *P*, and all these results show that there is no degradation in the final steady-state $MSE(n)$ of the reduced size predictor algorithm even for *P*<<*L*. The convergence speed and tracking capability of the reduced size predictor algorithm can be adjusted by changing the choice of the parameters λ, η and c_a.

4 Conclusion

We have developed algorithms for adaptive filtering. We have proposed a new (M-SMFTF) algorithm for adaptive filtering with fast convergence and low complexity. We have presented more complexity reduction of simplified FTF type algorithm by using a new recursive method to compute the likelihood variable. The computational complexity of the M-SMFTF algorithm is 6*L* operations per sample and this computational complexity can be significantly reduced to (2*L*+4*P*) when used with a reduced *P*-size (*P*<<*L*) forward predictor. This can be very interesting for long filters. The low computational complexity of the M-SMFTF when dealing with long filters and it a performance capabilities render it very interesting for applications such as acoustic echo cancellation. The proposed algorithm outperforms the classical adaptive algorithms because of its convergence speed which approaches that of the RLS algorithm and its computational complexity which is slightly greater than the one of the NLMS algorithm. The simulation has shown that the performances of proposed algorithm are better than those of the normalized least mean square algorithm.

Reference

[1] S.Haykin, *Adaptive Filter Theory*, 4th ed. Englewood Cliffs, NJ: Prentice-Hall, 2002.

[2] A.H.Sayed, *Fundamentals of Adaptive Filtering*, JohnWiley &Sons, New York, NJ, USA, 2003.

[3] O.Macchi, *The Least Mean Squares Approach with Applications in Transmission*, New York: Wiley, 1995.

[4] J.R.Treichler., C.R.Johnson and M.G.Larimore, *Theory and Design of Adaptive Filter*, Prentice Hall, 2001.

[5] L.Ljung, M.Morf and D.Falconner, Fast calculation of gain matrices for recursive estimation schemes, *Int. J. Control*., Vol.27, 1978, pp.1-19.

[6] G.Caryannis, D.Manolakis and N.Kalouptsidis, A fast sequential algorithm for least squares filtering and prediction, *IEEE Trans. Acoust. Speech Signal Process*, Vol.31, 1983, pp.1394-1402.

[7] J.Cioffi and T.Kailath, Fast RLS Transversal Filters for adaptive filtering, *IEEE press. On ASSP*, Vol.32, No.2, 1984, pp.304-337.

[8] A.Benallal and A.Gilloire, A New method to stabilize fast RLS algorithms based on a first-order model of the propagation of numerical errors, *Proc. ICASSP, New York*, USA, 1988, pp.1365-1368.

[9] D.T.M.Slock and T.Kailath, Numerically stable fast transversal filters for recursive least squares adaptive filtering, *IEEE transactions on signal processing*, Vol.39, No.1, 1991, pp.92–114.

[10] A.P.Liavas and P.A.Regalia, Numerical stability issues of the conventional recursive least squares algorithm, *Proc. ICASSP, Seattle*, WA, 1998.

[11] M.Arezki, A.Benallal, A.Guessoum, and D.Berkani, Three New Versions of Numerically Stable Fast Least Squares Algorithms (NS-FRLS) for Adaptive Filtering, *Proc. of the 4th international symposium on Communication Systems, Networks and Digital Signal Processing*, University of Newcastle, U.K, 2004, pp. 528-532.

[12] M.Arezki, A.Benallal, P.Meyrueis, A.Guessoum and D.Berkani, Analysis of Fast Recursive Least Squares Algorithms for Adaptive Filtering, *Proc. 11th WSEAS International Conference on SYSTEMS,* Crete Island, Greece, July 23-25, 2007, pp.474-481.

[13] S.Binde, A numerically stable fast transversal filter with leakage correction, *IEEE Signal Processing Letters*, Vol.2, No.6, 1995, pp.114–116.

[14] G.V.Moustakides, S.Theodoridis, Fast Newton transversal filters - A new class of adaptive estimation algorithms, *IEEE Trans. Signal Process*, Vol.39, No.10, 1991, pp.2184–2193.

[15] P.P.Mavridis and G.V.Moustakides, Simplified Newton-Type Adaptive Estimation Algorithms, *IEEE Trans. Signal Process*, Vol.44, No.8, 1996.

[16] W.Youhua, K.Ikeda and K.Nakayama, A numerically stable fast Newton-type adaptive filter based on order recursive least squares algorithm, *IEEE Transactions on Signal Processing*, Vol.51, No.9, 2003, pp.2357–2368.

[17] A.Benallal and A.Benkrid, A simplified FTF-type algorithm for adaptive filtering, *Signal processing*, Vol.87, No.5, 2007, pp.904–917.

[18] D.T.M.Slock, On the convergence behaviour of the LMS and the NLMS algorithms, *IEEE Trans. Signal Processing*, Vol.42, 1993, pp.2811–2825.

[19] A.Gilloire, E.Moulines, D.Slock, and P.Duhamel, *State of art in echo cancellation*, A.R. Figuers-vidal, Digital Signal processing in telecommunication, Springer, Berlin, 1996, pp.45–91