A New Algorithm with Low Complexity for Adaptive Filtering

M. Arezki, A. Benallal, P. Meyrueis and D. Berkani

***Abstract*—In this paper, we propose a new algorithm M-SMFTF 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. The computational complexity was reduced from 7*L* to 6*L*, where *L* is the finite impulse response filter length. Furthermore, this computational complexity can be significantly reduced to (2*L*+4*P*) when used with a reduced *P*-size forward predictor. 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. 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 normalized LMS algorithm.**

***Index Terms*— Fast RLS, NLMS, FNTF, Adaptive Filtering, Convergence Speed, Tracking capability.**

I. Introduction

There are two major classes of adaptive algorithms [1]-[3]. One is the least mean square (LMS) algorithm, which is based on a stochastic gradient method. The LMS algorithm has a computational complexity of $O(L)$, L is the finite impulse (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]-[12]. Another way of reducing the complexity of the fast RLS (FRLS) algorithm has been proposed in [13], [14]: 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. Recently, the simplified FTF-type algorithm [15] 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 7L when used with a full size predictor which is less complex than the original numerically stable 8L 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 6L and this computational complexity can be significantly reduced to ($2L$+4P) 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.

II. Adaptive Algorithms

The main identification block diagram of a linear system with finite impulse response (FIR) is represented in Fig.1. The output a priori error $\bar{\varepsilon}_{L,n}$ of this system at time n is:

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

Manuscript received December 19, 2008. This work was supported in part by the LATSI Laboratory of the Department of Electronics, University of Blida Algeria and the LSP Laboratory, University of Louis Pasteur, Strasbourg, France.

M. Arezki is with the Department of Electronics, University of Blida, B.P. 270 Route de Soumâa Blida 09000 Algeria. (corresponding author to provide phone: +213771204390; fax: +21325433850; e-mail: md_arezki@hotmail.com)

A. Benallal was with the Department of Electronics, University of Blida Algeria. He is now with the Department of Electronics, College of Technology, Dammam 7650, Saudi Arabia. (e-mail: a_benallal@hotmail.com)

P. Meyrueis is with the LSP Laboratory, University of Louis Pasteur, ENSPS, Bd. Sébastien Brant – B.P. 10413 ILLKIRCH 67412 France. (e-mail: meyruies@sphot.u-strasbg.fr).

D. Berkani, is with the Department of Electrical and Computer Engineering of ENP Algiers, 10 Avenue Pasteur, B.P. 182 El Harrach - 16000 Algiers Algeria. (e-mail: dberkani@hotmail.com).

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

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

where $\mathbf{w}_{\mathrm{opt},L} = [w_{\mathrm{opt},1}, \ldots, w_{\mathrm{opt},L}]^{\mathrm{T}}$ 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 ^{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.

A. 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 $\mu = 1$ [16]. The power $\pi_{x,n}$ of input signal can alternatively be estimated using the following recursive equation [17]:

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

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

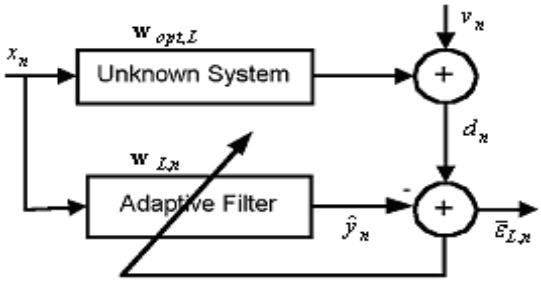

Fig.1: Main block diagram of an adaptive filter

B. 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<λ≤1). The adaptation gain is given by:

$$\mathbf{g}_{L,n} = \underbrace{\mathbf{R}_{L,n}^{-1} \mathbf{x}_{L,n}}_{\text{RLS}} = \underbrace{\gamma_{L,n} \tilde{\mathbf{k}}_{L,n}}_{\text{FRLS}} \tag{7}$$

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

where $\mathbf{R}_{L,n}$ is an estimate of the correlation matrix of the input signal vector. The variables $\gamma_{L,n}$ and $\tilde{\mathbf{k}}_{L,n}$ 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 fast RLS (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, also called 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

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

with $\bar{r}_{L,n}^{\mathrm{f}} = [(1-\mu_s)\bar{r}_{L,n}^{\mathrm{f}_0} + \mu_s \bar{r}_{L,n}^{\mathrm{f}_1}]$ and $0 \le \mu_s \le 1$, where ($\bar{r}_{L,n}$, $\bar{r}_{L,n}^{\mathrm{f}_0}$ and $\bar{r}_{L,n}^{\mathrm{f}_1}$) are the backward a priori prediction errors calculated in tree differently 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. 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]:

$$\lambda > 1 - 1/2L \tag{10}$$

The resulting stabilized FRLS (NS-FRLS) algorithms have a complexity of $8L$ as shown in Table I. Note that

numerical stabilization of the algorithm limits the range of the forgetting factor λ (10) and consequently their convergence speed and tracking ability.

III. Proposed Algorithms

A. The M-SMFTF Algorithm

We propose more complexity reduction of the simplified FTF-type (M-SMFTF) algorithm by using a new recursive method to compute the likelihood variable. The Simplified FTF-type algorithm [15] derived from the FTF algorithm where the adaptation gain is obtained only from the forward prediction variables. The backward prediction variables, which are the main source of the numerical instability in the FRLS algorithms, 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.

If we discard all backward prediction variables and use only the forward variables from the equation below:

$$\begin{bmatrix} \tilde{\mathbf{k}}_{L,n} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{\mathbf{k}}_{L,n-1} \end{bmatrix} + \frac{\bar{e}_{L,n}}{\lambda\alpha_{L,n-1}} \begin{bmatrix} 1 \\ -\mathbf{a}_{L,n-1} \end{bmatrix} - \frac{\bar{r}_{L,n}}{\lambda\beta_{L,n-1}} \begin{bmatrix} -\mathbf{b}_{L,n-1} \\ 1 \end{bmatrix} \tag{11}$$

the normalized Kalman gain can be rewritten as follows:

$$\begin{bmatrix} \tilde{\mathbf{k}}_{L,n} \\ * \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{\mathbf{k}}_{L,n-1} \end{bmatrix} + \frac{\bar{e}_{L,n}}{\lambda\alpha_{L,n-1}} \begin{bmatrix} 1 \\ -\mathbf{a}_{L,n-1} \end{bmatrix} \tag{12}$$

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

$$\frac{\bar{e}_{L,n}}{\lambda\alpha_{L,n-1}} \to \frac{\bar{e}_{L,n}}{\lambda\alpha_{L,n-1} + c_a} \tag{13}$$

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 $\mathbf{a}_{L,n}$

Table I: NS-FRLS algorithm (8*L*)

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

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

Variables available at the discrete-time index n:

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

New information: x_n , d_n .

- Prediction Part:

Modeling of x_n , x_{n-L}

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

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

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

$\bar{r}_{L,n} = x_{n-L} - \mathbf{b}_{L,n-1}^{\mathrm{T}} \mathbf{x}_{L,n}$;

$\bar{r}_{L,n}^{\mathrm{f}0} = \lambda\beta_{L,n-1}\tilde{k}_{L+1,n}^{+}$; $\bar{r}_{L,n}^{\mathrm{f}1} = \lambda^{-L+1}\gamma_{L,n-1}\alpha_{L,n-1}\tilde{k}_{L+1,n}^{+}$

$\xi_n = \bar{r}_{L,n} - [(1-\mu_s)\bar{r}_{L,n}^{\mathrm{f}0} + \mu_s \bar{r}_{L,n}^{\mathrm{f}1}]$;

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

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

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

- Filtering Part:

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

return back to zero by applying the following operation:

$$\mathbf{a}_{L,n} \to \eta\,\mathbf{a}_{L,n} \tag{14}$$

where η is a constant close to one often called the leakage factor [12]. The likelihood variable is given by using the definition directly:

$$\gamma_{L,n} = \frac{1}{1 + \tilde{\mathbf{k}}_{L,n}^{\mathrm{T}}\mathbf{x}_{L,n}} \tag{15}$$

Let us replace the quantity (*), that has not been used in $\tilde{\mathbf{k}}_{L,n}$ of (12), by the variable $c_{L,n}$, we obtain:

$$\begin{bmatrix} \tilde{\mathbf{k}}_{L,n} \\ c_{L,n} \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{\mathbf{k}}_{L,n-1} \end{bmatrix} + \frac{\bar{e}_{L,n}}{\lambda\alpha_{L,n-1} + c_a} \begin{bmatrix} 1 \\ -\mathbf{a}_{L,n-1} \end{bmatrix} \tag{16}$$

We can write the input signal vector extended to the order (*L*+1) in two different ways as shown below:

$$\mathbf{x}_{L+1,n} = \left[\mathbf{x}_{L,n}^{\mathrm{T}},\ x_{n-L}\right]^{\mathrm{T}} \quad (17a); \qquad \mathbf{x}_{L+1,n} = \left[x_n,\ \mathbf{x}_{L,n-1}^{\mathrm{T}}\right]^{\mathrm{T}} \quad (17b)$$

If we multiply on the left, the members of left and right of expression (16) by equations (17a) and (17b) respectively; we get:

$$\mathbf{x}_{L,n}^{\mathrm{T}}\tilde{\mathbf{k}}_{L,n} + c_{L,n}\,x_{n-L} = \mathbf{x}_{L,n-1}^{\mathrm{T}}\tilde{\mathbf{k}}_{L,n-1} + \frac{\bar{e}_{L,n}^2}{\lambda\alpha_{L,n-1} + c_a} \tag{18}$$

By rearranging (18), we obtain a new recursive formula for calculating the likelihood variable which is much less

Table II: M-SMFTF Algorithm (6*L*)

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:**

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

complex than (15), the new expression is given below:

$$\gamma_{L,n} = \frac{\gamma_{L,n-1}}{1 + \delta_{L,n} \gamma_{L,n-1}} \tag{19}$$

$$\delta_{L,n} = \frac{\bar{e}_{L,n}^2}{\lambda \alpha_{L,n-1} + c_a} - c_{L,n}\, x_{n-L} \tag{20}$$

The computational complexity of the M-SMFTF algorithm is 6*L*; as illustrated in Table II.

B. The RM-SMFTF Algorithm

The Reduced size predictors in the FTF algorithms have been successfully used in the FNTF algorithms [13]-[15]. The M-SMFTF 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:

$$\mathbf{a}_{L,n} = \begin{bmatrix} \mathbf{a}_{P,n} \\ 0_{L-P} \end{bmatrix} \tag{21}$$

$$\begin{bmatrix} \widetilde{\mathbf{k}}_{L,n} \\ c_{L,n} \end{bmatrix} = \begin{bmatrix} 0 \\ \widetilde{\mathbf{k}}_{L,n-1} \end{bmatrix} + \frac{\bar{e}_{P,n}}{\lambda \alpha_{P,n-1} + c_a} \begin{bmatrix} 1 \\ -\mathbf{a}_{P,n-1} \\ 0_{L-P} \end{bmatrix} \tag{22}$$

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 $\alpha_{P,n}$, 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 $\mathbf{w}_{L,n}$. The computational complexity of the reduced size M-SMFTF (RM-SMFTF) algorithm is (2*L*+4*P*); it is given in Table III.

Table III: RM-SMFTF Algorithm (2*L*+4*P*; *P*<<*L*)

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**:

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

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

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

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

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

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

- **Filtering Part:**

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

C. Convergence analysis

The analysis uses the common independence assumption that the current input signal vector is statistically independent of the current coefficient vector of the adaptive filter. We define the weight-error vector at time *n* as:

$$\Delta \mathbf{w}_{L,n} = \mathbf{w}_{\text{opt},L} - \mathbf{w}_{L,n} \tag{23}$$

The output a priori error $\bar{\varepsilon}_{L,n}$ can be written as:

$$\bar{\varepsilon}_{L,n} = v_n + \mathbf{x}_{L,n}^{\mathrm{T}} \Delta \mathbf{w}_{L,n-1} \tag{24}$$

The recursion in (3) on the coefficient error vector is:

$$\Delta \mathbf{w}_{L,n} = [\mathbf{I}_L - \gamma_{L,n} \widetilde{\mathbf{k}}_{L,n} \mathbf{x}_{L,n}^{\mathrm{T}}] \Delta \mathbf{w}_{L,n-1} - \gamma_{L,n} \widetilde{\mathbf{k}}_{L,n} v_n \tag{25}$$

In a steady state operation, we can write:

$$\alpha_{L,n-i} \to \alpha(\infty) = \gamma(\infty) \frac{\sigma_x^2}{1-\lambda} \tag{26}$$

where $\gamma(\infty)$ is asymptotic value of the slowly variable $\gamma_{L,n}$. The normalized Kalman gain becomes:

$$\widetilde{\mathbf{k}}_{L,n} = \frac{1}{\lambda \alpha(\infty)} \mathbf{x}_{L,n} \tag{27}$$

Replacing the variable $\gamma_{L,n}$ by its steady-state value $\gamma(\infty)$ and using (26) and (27), the recursive expression (23) becomes:

$$\Delta \mathbf{w}_{L,n} = \left[\mathbf{I}_L - \frac{(1-\lambda)}{\lambda \sigma_x^2} \mathbf{x}_{L,n} \mathbf{x}_{L,n}^{\mathrm{T}} \right] \Delta \mathbf{w}_{L,n-1} - \frac{(1-\lambda)}{\lambda \sigma_x^2} v_n \tag{28}$$

By taking the expected value of both sides of (28) and using the independence assumption to yield:

$$\mathrm{E}\{\Delta\mathbf{w}_{L,n}\} = (2-\lambda^{-1})\,\mathrm{E}\{\Delta\mathbf{w}_{L,n-1}\} \tag{29}$$

The steady-state solution of (29) is: If $\lambda < 1 \Rightarrow \mathrm{E}\{\Delta\mathbf{w}(\infty)\} = 0_L$; from which we obtain the steady-state mean coefficient vector of the M-SMFTF adaptive filter as:

$$\mathrm{E}\{\mathbf{w}(\infty)\} = \mathbf{w}_{\mathrm{opt},L} \tag{30}$$

D. Analysis Prediction Part

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

$$\Delta\boldsymbol{\varphi}_n = \mathbf{F}(n)\,\Delta\boldsymbol{\varphi}_{n-1} + \mathbf{e}(n) \tag{31}$$

where $\mathbf{e}(n)$ represent the round-off noise. We can write the state vector of the errors at the time n as follows:

$$\Delta\boldsymbol{\varphi}_n = \left[\Delta\boldsymbol{\varphi}_n^{a\,\mathrm{T}} \;\; \Delta\boldsymbol{\varphi}_n^{k\,\mathrm{T}}\right]^{\mathrm{T}} \tag{32}$$

where $\Delta\boldsymbol{\varphi}_n^a = \left(\Delta\mathbf{a}_{L,n}^{\mathrm{T}},\, \Delta\alpha_{L,n}\right)^{\mathrm{T}}$ and $\Delta\boldsymbol{\varphi}_n^k = \left(\Delta\widetilde{\mathbf{k}}_{L,n}^{\mathrm{T}}, \Delta\gamma_{L,n}\right)^{\mathrm{T}}$ represent respectively the errors cumulated up until the time n in the forward and Kalman recursive variables. The $(2L+2)$x$(2L+2)$ dimensional matrix $\mathbf{F}(n)$ given by:

$$\mathbf{F}(n) = \begin{bmatrix} \mathbf{F}_{11}(n) & \mathbf{F}_{12}(n) \\ \mathbf{F}_{21}(n) & \mathbf{F}_{22}(n) \end{bmatrix} \tag{33}$$

represents the transition matrix. The system (31) 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. After a propagation analysis of the numerical errors of the 1st order and an asymptotic study of the equations of errors propagation, we approximate the errors in the forward variables ($\Delta\mathbf{a}_{L,n}^{\mathrm{T}}$, $\Delta\alpha_{L,n}$) and the Kalman variables ($\Delta\widetilde{\mathbf{k}}_{L,n}^{\mathrm{T}}$, $\Delta\gamma_{L,n}$) by the following linear first order models deduced from differentiating ($\mathbf{a}_{L,n}^{\mathrm{T}}, \alpha_{L,n}$) and ($\widetilde{\mathbf{k}}_{L,n}^{\mathrm{T}}, \gamma_{L,n}$) respectively:

$$\Delta\mathbf{a}_{L,n} = \eta\left(\mathbf{I}_L - \gamma_{L,n-1}\widetilde{\mathbf{k}}_{L,n-1}\mathbf{x}_{L,n-1}^{\mathrm{T}}\right)\Delta\mathbf{a}_{L,n-1} + \mathbf{p}_a(n) \tag{34a}$$

$$\Delta\alpha_{L,n} = \lambda\,\Delta\alpha_{L,n-1} + p_\alpha(n) \tag{34b}$$

$$\Delta\widetilde{\mathbf{k}}_{L,n} = \mathbf{M}^k\,\Delta\widetilde{\mathbf{k}}_{L,n-1} + \mathbf{p}_k(n) \tag{35a}$$

$$\Delta\gamma_{L,n} = c^\gamma(n)\,\Delta\gamma_{L,n-1} + p_\gamma(n) \tag{35b}$$

with

$$\mathbf{M}^k = \begin{bmatrix} 0_{L-1}^{\mathrm{T}} & 0 \\ \mathbf{I}_{L-1} & 0_{L-1} \end{bmatrix} \tag{36}$$

$$c^\gamma(n) = \frac{\gamma_{L,n}}{\gamma_{L,n-1}}(1-\delta_{L,n}\gamma_{L,n}) \tag{37}$$

By assuming that the perturbation terms ($\mathbf{p}_a(n)$, $p_\alpha(n)$, $\mathbf{p}_k(n)$, $p_\gamma(n)$) remain limited. In an asymptotic mode, we can write:

$$\eta\left(\mathbf{I}_L - \gamma_{L,n-1}\widetilde{\mathbf{k}}_{L,n-1}\mathbf{x}_{L,n-1}^{\mathrm{T}}\right) \rightarrow \;\; \eta\,\lambda\,\mathbf{I}_L \tag{38}$$

$$\mathrm{E}\{c^\gamma(n)\} \rightarrow \frac{1}{\lambda^{-1}+\lambda-1} \tag{39}$$

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. To calculate the covariance matrix of numerical errors in forward predictor; we use a statistical approach, $\mathbf{A}_n = \mathrm{E}\{\Delta\mathbf{a}_{L,n}\Delta\mathbf{a}_{L,n}^{\mathrm{T}}\}$. 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. With an input signal Gaussian sequence, we obtain the following expressions:

$$\mathbf{A}_n = G\;\mathbf{A}_{n-1} + \mathrm{E}\{\mathbf{p}_a(n)\mathbf{p}_a^{\mathrm{T}}(n)\} \tag{40}$$

$$G = \eta^2\left(1-2(1-\lambda)+(1-\lambda)^2(L+2)\right) \tag{41}$$

The stability condition of equation (40) is given by the solution of the following inequality:

$$|G| < 1 \tag{42}$$

This inequality is only verified for this condition:

$$\lambda > 1 - \frac{1+\sqrt{1+\left(\frac{1}{\eta^2}-1\right)(L+2)}}{(L+2)} \approx 1-1/L \tag{43}$$

We note that the lower bound of this condition is always smaller than the lower bound of condition (10) of the original numerically stable FRLS algorithm. This 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.

IV. 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.

A. *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 (M-SMFTF, RM-SMFTF) algorithms with NS-FRLS and NLMS algorithms.

The filter length is L=256, the NLMS (μ=1) and the NS-FRLS ($\lambda = 1-1/3L$) algorithms are tuned to obtain fastest convergence. The forgetting factor λ and the leakage factor η for the proposed algorithms are chosen according to (43), $\lambda = 1-1/L$ of the M-SMFTF and $\lambda = 1-1/P$ of the RM-SMFTF algorithms, where P is the predictor order.

The nonstationarity of the system to be modelled is simulated by introducing a linear gain variation on the desired signal.

Fig.2 shows that better performances in convergence speed are obtained for the proposed algorithm. The differences in the final $MSE(n)$ for the proposed algorithms and NS-FRLS algorithm 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.

B. *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 algorithms with NS-FRLS and NLMS algorithms. We simulated an abrupt change in the impulse response at the 56320th samples. 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 $\lambda = 1-1/L$ and for the RM-SMFTF algorithm the forgetting factor is $\lambda = 1-1/P$, but the leakage η and the constant c_a must be carefully chosen.

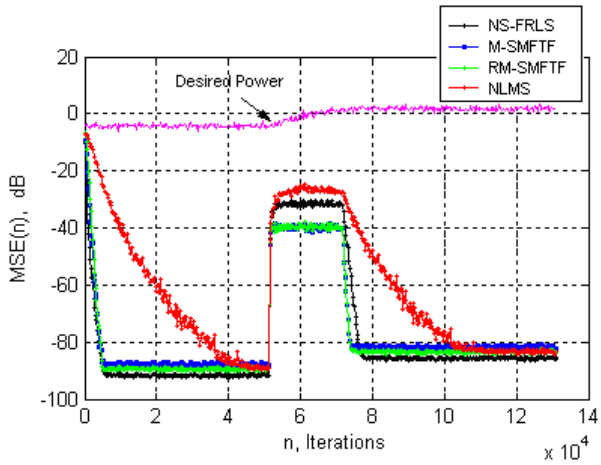

Fig.2: Comparative performance of the algorithms for USASI noise, L=256. M-SMFTF: λ=0.9961, η=0.985, c_a=0.5, E_0=1; RM-SMFTF: P=32, λ=0.9688, η=0.9985, c_a=0.5, E_0=0.2; NS-FRLS: λ=0.9987, E_0=1; NLMS: μ=1

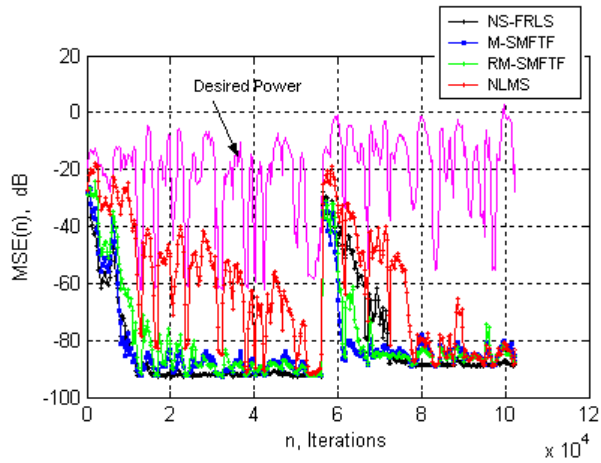

Fig.3: Comparative performance of the algorithms with speech input, L=256. M-SMFTF: λ=0.9961, η=0.96, c_a=0.1, E_0=0.5; RM-SMFTF: P=32, λ=0.9688, η=0.99, c_a=0.1, E_0=0.1; NS-FRLS: λ=0.9996, E_0=0.5; NLMS: μ=1.

In Fig.3, we can see that the initial convergence is almost the same for both M-SMFTF and NS-FRLS algorithms. However the proposed algorithm achieves better re-convergence after the abrupt change in the impulse response.

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.

V. Conclusion

We have presented a new algorithm M-SMFTF for adaptive filtering with fast convergence and low complexity. We have proposed 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 $6L$ operations per sample and this computational complexity can be significantly reduced to ($2L$+$4P$) 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 its performance capabilities makes it very interesting for applications such as acoustic echo cancellation. The simulation has shown that the performances of the proposed algorithm are better than those of the normalized least mean square algorithm.

References

[1] S. Haykin, *Adaptive Filter Theory*, 4th ed. ,NJ, Prentice-Hall, 2002.
[2] A.H. Sayed, *Fundamentals of Adaptive Filtering*, John Wiley & 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, P. Meyrueis, A. Guessoum and D. Berkani, "Error Propagation Analysis of Fast Recursive Least Squares Algorithms," *Proc. 9th IASTED International Conference on Signal and Image Processing*, Honolulu, Hawaii, USA, August 20–22, 2007, pp.97-101.
[12] S. Binde, "A numerically stable fast transversal filter with leakage correction," *IEEE Signal Processing Letters*, 1995, 2, (6), pp.114-116.
[13] G.V. Moustakides and 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.
[14] P. P. Mavridis and G. V. Moustakides, "Simplified Newton-Type Adaptive Estimation Algorithms," *IEEE Trans. Signal Process*, vol.44, no.8, 1996.
[15] A. Benallal and A. Benkrid, "A simplified FTF-type algorithm for adaptive filtering," *Signal processing*, vol.87, no.5, 2007, pp.904-917.
[16] 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.
[17] 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

M. Arezki has received the degree of Engineer in electronics from the "Ecole Nationale Polytechique" ENP, Algiers, Algeria, in 1986. He obtained a Master in Electrical Engineering of university of Blida, Algeria in 1997.
He has been with the University of Blida, Algeria, as a teacher Assistant, in the department of Electronics since 1989 and a Research Assistant in the Digital Signal and Image Processing Research Laboratory since 2000. His current research interests include adaptive filtering, acoustic echo cancellation, identification, fast algorithms.