IMPROVEMENT OF THE SIMPLIFIED FTF-TYPE ALGORITHM

Madjid Arezki, Ahmed Benallal, Abderezak Guessoum
LATSI Laboratory - Department of Electronics, Faculty of Engineering-University of Blida, Algeria

Daoud Berkani
Signal & Communications Laboratory - (ENP), Algiers, Algeria

Keywords: Fast RLS, NLMS, FNTF, Adaptive Filtering, Convergence Speed, Tracking capability.

Abstract: In this paper, we propose a new algorithm M-SMFTF which reduces the complexity of the simplified FTF-type (SMFTF) algorithm by 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. Finally, some simulation results are presented and our algorithm shows an improvement in convergence over the normalized least mean square (NLMS).

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 (Haykin, 2002) and (Sayed, 2003). There are two major classes of adaptive algorithms. One is the least mean square (LMS) algorithm, which is based on a stochastic gradient method (Macchi, 1995). Its computational complexity is 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 (Treichler, 2001). 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 (Treichler, 2001). The fast transversal filter (FTF) (Cioffi, 1984) algorithm is derived from the RLS by the introduction of forward and backward predictors. Its computational complexity is of $O(L)$. Several numerical solutions of stabilization, with stationary signals, are proposed in the literature (Benallal, 1988), (Slock, 1991) and (Arezki, 2007). Another way of reducing the complexity of the fast RLS algorithm has been proposed in (Moustakides, 1999) and (Mavridis, 1996). 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 (SMFTF) algorithm (Benallal, 2007) 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 the SMFTF algorithm is 7L. 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. At the end, we present some simulation results of the M-SMFTF algorithm.

Arezki M., Benallal A., Guessoum A. and Berkani D. (2008).
IMPROVEMENT OF THE SIMPLIFIED FTF-TYPE ALGORITHM.
In *Proceedings of the International Conference on Signal Processing and Multimedia Applications*, pages 156-161
DOI: 10.5220/0001940001560161

2 ADAPTIVE ALGORITHMS

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

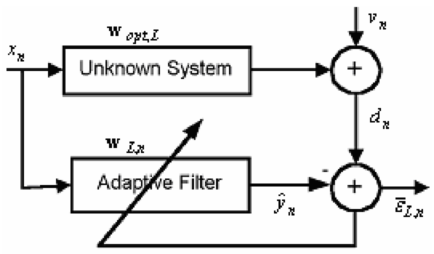

Figure 1: Main block diagram of an adaptive filter.

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

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

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

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

where $\mathbf{w}_{opt,L} = [w_{opt,1}, w_{opt,2}, \ ..., w_{opt,L}]^{\mathrm{T}}$ 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

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

The different algorithms are distinguished by the gain calculation.

2.1 The NLMS Algorithm

The Algorithms derived from the gradient (Macchi, 1995), 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:

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

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 $\mu = 1$ (Slock, 1993). The power of input signal $\pi_{x,n}$ is given by:

$$\pi_{x,n} = \frac{\mathbf{x}^{\mathrm{T}}_{L,n}\mathbf{x}_{L,n}}{L} \quad (5a)$$

It can alternatively be estimated using following recursive equation (Gilloire, 1996):

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

where γ is a forgetting factor ($\gamma \approx 1/L$). The computational complexity of the NLMS algorithm is $2L$ multiplications per sample for the version with the recursive estimator (5b).

2.2 The SFRLS Algorithm

The filter $\mathbf{w}_{L,n}$ is calculated by minimizing the weighted least squares criterion according to (Haykin, 2002):

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

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

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

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. The calculation complexity of a FRLS algorithm is $7L$. This reduction of complexity has made all FRLS algorithms numerically unstable. The numerical stability is obtained by using some redundant formulae of the FRLS algorithms (Benallal, 1988), (Slock, 1991) and (Arezki, 2007). The numerical stability is obtained by using a control variable, called also a divergence indicator ξ_n (Arezki, 2007), theoretically equals to zero. Its introduction in an unspecified point of the algorithm modifies its numerical properties. This variable is given by:

$$\xi_n = \bar{r}_{L,n} - \bar{r}^{\mathrm{f}}_{L,n} \begin{cases} = 0 & \text{theory} \\ \neq 0 & \text{practical} \end{cases} \quad (8)$$

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 calculated differently in tree ways.

We define three backward a priori prediction errors ($\bar{r}^{\gamma}_{L,n}, \bar{r}^{\beta}_{L,n}, \bar{r}^{\mathrm{b}}_{L,n}$), 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:

$$\mu^{\gamma} = 0, \;\; \mu^{\beta} = \mu^{b} = 1; \;\; \mu_s = 0.5 \tag{9}$$

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 (Arezki, 2007):

$$\lambda > \frac{4L+5}{4L+7} = 1 - \frac{1}{2L+3.5} \tag{10}$$

These conditions can be written in another simpler form $\lambda = 1 - 1/pL$, where the parameter p is a real number strictly greater than 2 to ensure numerical stability. The resulting stabilized FRLS (SFRLS) algorithms have a complexity of 8*L*; it is given in Table 1. Note that numerical stabilization of the algorithm limits the range of the forgetting factor λ (condition (10)) and consequently their convergence speed and tracking ability.

Table 1: SFRLS (8*L*) algorithm.

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

Modeling of x_n , x_{n-L}

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

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

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

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

$\bar{r}^{\mathrm{f0}}_{L,n} = \lambda \beta_{L,n-1} \tilde{k}^{+}_{L+1,n}$; $\bar{r}^{\mathrm{f1}}_{L,n} = \lambda^{-L+1} \gamma_{L,n-1} \alpha_{L,n-1} \tilde{k}^{+}_{L+1,n}$

$\xi_n = \bar{r}_{L,n} - [(1-\mu_s) \bar{r}^{\mathrm{f0}}_{L,n} + \mu_s \bar{r}^{\mathrm{f1}}_{L,n}]$;

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

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

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

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

2.3 The SMFTF Algorithm

The Simplified FTF-type (SMFTF) algorithm (Benallal, 2007) 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 (Benallal, 1988), (Slock, 1991) and (Arezki, 2007), 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.

Discarding the backward predictor does not mean that the last components $\mathbf{w}_{L,n}$ are not updated, but they are updated by components coming from lower positions of $\tilde{\mathbf{k}}_{L,n}$. To avoid the instability of the algorithm, we append a small positive constant c_a to the denominator $\bar{e}_{L,n} / (\lambda \alpha_{L,n-1} + c_a)$, and it might be preferable to have the forward predictor $\mathbf{a}_{L,n}$ return back to zero by doing $\eta \, \mathbf{a}_{L,n}$, where η is a close to one constant (Slock, 1993). The computational complexity of the SMFTF algorithm is 7*L*; it is given in Table 2.

Table 2: SMFTF (7*L*) algorithm.

Initialization:

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

Variables available at the discrete-time index *n*:

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

New information: x_n , d_n

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

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

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

$\gamma_{L,n} = \dfrac{1}{1 + \tilde{\mathbf{k}}^{\mathrm{T}}_{L,n} \mathbf{x}_{L,n}}$;

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

3 PROPOSED ALGORITHMS

3.1 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. Let us replace the quantity (*), that has not been used in $\tilde{\mathbf{k}}_{L,n}$ of the MSFTF algorithm (Table 2), by the variable $c_{L,n}$, we obtain:

$$\begin{bmatrix} \tilde{\mathbf{k}}_{L,n} \\ c_{L,n} \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{\mathbf{k}}_{L,n-1} \end{bmatrix} + \frac{\bar{e}_{L,n}}{\lambda\alpha_{L,n-1} + c_a} \begin{bmatrix} 1 \\ -\mathbf{a}_{L,n-1} \end{bmatrix} \quad (11)$$

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:

$$\mathbf{x}_{L+1,n} = \left[\mathbf{x}_{L,n}^{\mathrm{T}},\ x_{n-L}\right]^{\mathrm{T}} \quad (12a)$$

$$\mathbf{x}_{L+1,n} = \left[x_n,\ \mathbf{x}_{L,n-1}^{\mathrm{T}}\right]^{\mathrm{T}} \quad (12b)$$

By multiplying on the left, the members of left and right of the expression (11) by equations (12a) and (12b) respectively, the following equality is obtained:

$$\mathbf{x}_{L,n}^{\mathrm{T}}\tilde{\mathbf{k}}_{L,n} + c_{L,n}\, x_{n-L} = \mathbf{x}_{L,n-1}^{\mathrm{T}}\tilde{\mathbf{k}}_{L,n-1} + \frac{\bar{e}_{L,n}^2}{\lambda\alpha_{L,n-1} + c_a} \quad (13)$$

By manipulating the relation (13), we obtain a new recursive formula for calculating the likelihood variable as given below:

$$\gamma_{L,n} = \frac{\gamma_{L,n-1}}{1 + \delta_{L,n}\gamma_{L,n-1}} \quad (14)$$

$$\delta_{L,n} = \frac{\bar{e}_{L,n}^2}{\lambda\alpha_{L,n-1} + c_a} - c_{L,n}\, x_{n-L} \quad (15)$$

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 ($\Delta\mathbf{a}_{L,n}, \Delta\alpha_{L,n}$) and the Kalman variables ($\Delta\tilde{\mathbf{k}}_{L,n-1}, \Delta\gamma_{L,n-1}$) by the linear first order models deduced from differentiating ($\mathbf{a}_{L,n}, \alpha_{L,n}$) and ($\tilde{\mathbf{k}}_{L,n}, \gamma_{L,n}$) respectively. We can thus say that the system is numerically stable, in the mean sense, for λ and η between zero and one. It can be shown that the variance of the numerical errors in the forward predictor, with the assumption of a white Gaussian input signal, is stable under the following condition:

$$\lambda > 1 - \frac{1 + \sqrt{1 + \left(\frac{1}{\eta^2} - 1\right)(L+2)}}{(L+2)} \quad (16)$$

We notice that the lower bound of this condition is always smaller than the lower bound of condition (10) 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. The computational complexity of the M-SMFTF algorithm is $6L$; it is given in Table 3.

Table 3: M-SMFTF ($6L$) algorithm.

Initialization:
$\mathbf{w}_{L,0} = \mathbf{a}_{L,0} = \tilde{\mathbf{k}}_{L,0} = 0_L$; $\gamma_{L,0} = 1$; $\alpha_{L,0} = \lambda^L E_0$; ; $E_0 \geq \sigma_x^2 L/100$
Variables available at the discrete-time index n:
$\mathbf{a}_{L,n-1}; \tilde{\mathbf{k}}_{L,n-1}; \gamma_{L,n-1}; \alpha_{L,n-1}; \mathbf{w}_{L,n-1}$
New information: x_n , d_n .
$\bar{e}_{L,n} = x_n - \mathbf{a}_{L,n-1}^{\mathrm{T}}\mathbf{x}_{L,n-1}$;
$\begin{bmatrix} \tilde{\mathbf{k}}_{L,n} \\ c_{L,n} \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{\mathbf{k}}_{L,n-1} \end{bmatrix} + \frac{\bar{e}_{L,n}}{\lambda\alpha_{L,n-1} + c_a} \begin{bmatrix} 1 \\ -\mathbf{a}_{L,n-1} \end{bmatrix}$;
$\mathbf{a}_{L,n} = \eta\left\{\mathbf{a}_{L,n-1} + \bar{e}_{L,n}\gamma_{L,n-1}\tilde{\mathbf{k}}_{L,n-1}\right\}$; $\alpha_{L,n} = \lambda\alpha_{L,n-1} + \gamma_{L,n-1}\bar{e}_{L,n}^2$
$\delta_{L,n} = \frac{\bar{e}_{L,n}^2}{\lambda\alpha_{L,n-1} + c_a} - c_{L,n}\, x_{n-L}$; $\gamma_{L,n} = \frac{\gamma_{L,n-1}}{1 + \delta_{L,n}\gamma_{L,n-1}}$
$\bar{\varepsilon}_{L,n} = d_n - \mathbf{w}_{L,n-1}^{\mathrm{T}}\mathbf{x}_{L,n}$; $\mathbf{w}_{L,n} = \mathbf{w}_{L,n-1} + \bar{\varepsilon}_{L,n}\gamma_{L,n}\tilde{\mathbf{k}}_{L,n}$

3.2 The Reduced M-SMFTF Algorithm

The Reduced size predictors in the FTF algorithms have been successfully used in the FNTF algorithms (Moustakides, 1999), (Mavridis, 1996) and (Benallal, 2007). 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 normalized Kalman gain is given by:

$$\begin{bmatrix} \tilde{\mathbf{k}}_{L,n} \\ c_{L,n} \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{\mathbf{k}}_{L,n-1} \end{bmatrix} + \frac{\bar{e}_{P,n}}{\lambda\alpha_{P,n-1} + c_a} \begin{bmatrix} 1 \\ -\mathbf{a}_{P,n-1} \\ 0_{L-P} \end{bmatrix} \quad (17)$$

where P is much smaller than L. The first (P+1) components of the $\tilde{\mathbf{k}}_{L,n}$ are updated using the reduced size forward variables, the last components are just a shifted version of the (P+1)th component of $\tilde{\mathbf{k}}_{L,n}$. 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)th component of $\tilde{\mathbf{k}}_{L,n}$. 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 this algorithm is ($2L$+$4P$); it is given in Table 4.

Table 4: Reduced M-SMFTF (2*L*+4*P*) algorithm.

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

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

Variables available at the discrete-time index n:

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

New information: x_n , d_n .

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

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

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

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

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

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

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

4 SIMULATION RESULTS

To confirm the validity of our analysis and demonstrate the improved numerical performance, some simulations are carried out. 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 (16) with the stationary input. In our experiments, we have used values of c_a comparable with the input signal power.

4.1 The M-SMFTF Case

We define the norm gain-error by $NGE(n)$. This variable is used in our simulations to check the equality of the expressions of the likelihood variables. It is calculated by:

$$NGE(n) = 10 \log_{10} \left(\mathrm{E} \left\{ \left\| \Delta \mathbf{g}_{L,n} \right\|^2 \right\} \right) \tag{18}$$

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 SMFTF and M-SMFTF algorithms 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 SMFTF and M-SMFTF algorithms with a forgetting factor ($\lambda \geq (1-1/L)$) λ=0.9688, the leakage factor η=0.98 and c_a=0.1. In Figure 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. The M-SMFTF and the SMFTF algorithms produce exactly the same filtering error signal.

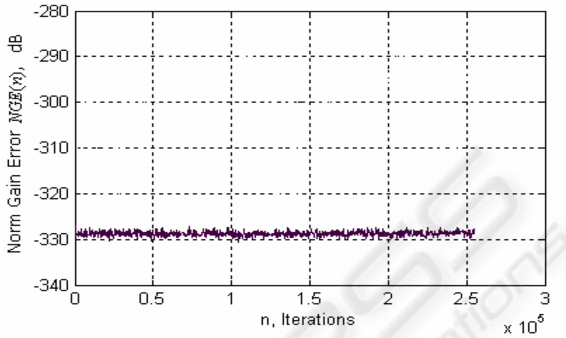

Figure 2: Evolution of the norm gain-error $NGE(n)$; L=32, λ=0.9688, η=0.98, c_a=0.1, E_0=0.5.

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 M-SMFTF algorithm with SFRLS and NLMS algorithms. The NLMS (μ=1) and SFRLS ($\lambda = 1 - 1/3L$) algorithms are tuned to obtain fastest convergence. We simulated an abrupt change in the impulse response by multiplying the desired signal by 1.5 in the steady state at the 51200^{th} samples. Figure 3 shows that better performances in convergence speed are obtained for the M-SMFTF algorithm.

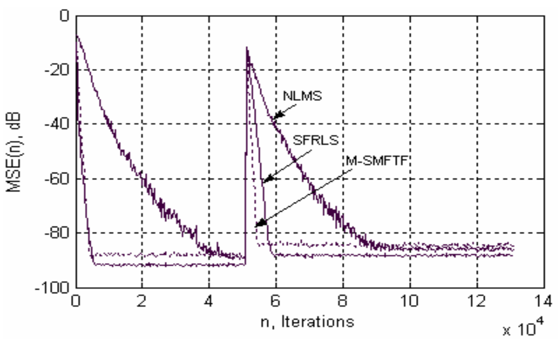

Figure 3: Comparative performance of the M-SMFTF, SFRLS and NLMS for USASI noise, L=256, M-SMFTF: λ=0.9961, η=0.985, c_a=0.5, E_0=1; SFRLS: λ=0.9987; NLMS: μ=1.

The differences in the final $MSE(n)$ for the M-SMFTF and SFRLS algorithms are due to the use of different forgetting factors λ .

4.2 The Reduced M-SMFTF Case

In this simulation, we compare the convergence performance of reduced size predictor M-SMFTF algorithm and the NLMS algorithm. Figure 4 presents the results obtained with the speech signal, sampled at 16 kHz, for the filter order L=256. We simulated an abrupt change in the impulse response at the 56320th samples. We use the following parameters: the predictor order is P=20, the forgetting factor is $\lambda = 1 - 1/P$. From this plot, we observe that the re-convergence of M-SMFTF is again faster than NLMS.

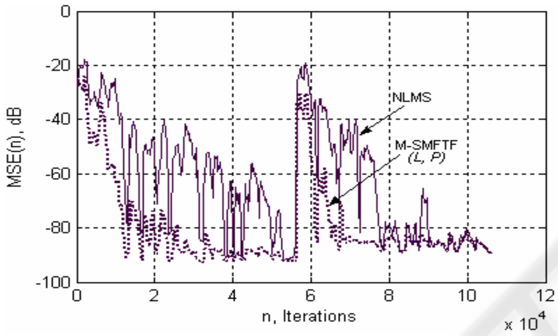

Figure 4: Comparative performance of the M-SMFTF and NLMS with speech input, L=256, M-SMFTF: P=20, λ=0.950, η=0.99, c_a=0.1, E_0=1; NLMS: μ=1.

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 .

5 CONCLUSIONS

We have proposed more complexity reduction of SMFTF (M-SMFTF) 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 forward predictor (P<<L). 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 simulation has shown that the performances of M-SMFTF algorithm are better than those of NLMS algorithm. The M-SMFTF 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.

REFERENCES

Arezki, M., Benallal, A., Meyrueis, P., Guessoum A., Berkani, D., 2007. 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, pp.97-101.

Benallal, A., Gilloire, A., 1988. 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, pp.1365-1368

Benallal, A., Benkrid, A., 2007. A simplified FTF-type algorithm for adaptive filtering. Signal processing, vol.87, no.5, pp.904-917.

Cioffi, J., Kailath, T., 1984. Fast RLS Transversal Filters for adaptive filtering. IEEE press. On ASSP, vol.32, no.2, pp.304-337.

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

Haykin, S., 2002. *Adaptive Filter Theory*, Prentice-Hall. NJ, 4th edition.

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

Mavridis, P.P., Moustakides, G.V., 1996. Simplified Newton-Type Adaptive Estimation Algorithms. IEEE Trans. Signal Process, vol.44, no.8.

Moustakides, G.V., Theodoridis, S., 1999. Fast Newton transversal filters - A new class of adaptive estimation algorithms. IEEE Trans. Signal Process, vol.39, no.10, pp.2184–2193.

Sayed, A.H., 2003. *Fundamentals of Adaptive Filtering*, John Wiley & Sons. NJ,

Slock, D.T.M., Kailath, T., 1991. Numerically stable fast transversal filters for recursive least squares adaptive filtering," IEEE transactions on signal processing, vol.39, no.1, pp.92-114.

Slock, D.T.M., 1993. On the convergence behaviour of the LMS and the NLMS algorithms. IEEE Trans. Signal Processing, vol.42, pp.2811-2825.

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