

Improvement of the Simplified Fast Transversal Filter Type Algorithm for Adaptive Filtering

¹Madjid Arezki, ²Ahmed Benallal and ³Daoud Berkani

¹LATSI Laboratory, Department of Electronic, University of Blida, Algeria

²Department of Electronics, Dammam College of Technology, Dammam 7650, Saudi Arabia

³Signal and Communications Laboratory, ENP, Algiers, Algeria

Abstract: Problem statement: In this study, we proposed a new algorithm M-SMFTF for adaptive filtering with fast convergence and low complexity. **Approach:** It was the result of a simplified FTF type algorithm, where the adaptation gain was obtained only from the forward prediction variables and using a new recursive method to compute the likelihood variable. **Results:** The computational complexity was reduced from $7L-6L$, where L is the finite impulse response filter length. Furthermore, this computational complexity can be significantly reduced to $(2L+4P)$ when used with a reduced P -size forward predictor. **Conclusion:** This algorithm presented 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.

Key words: Fast RLS, NLMS, FNTF, adaptive filtering, convergence speed, tracking capability

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^[8-12]. Another way of reducing the complexity of the 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 ($P \ll L$). 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.

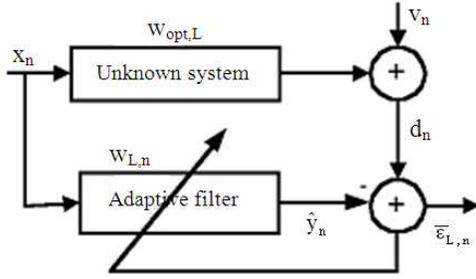


Fig.1: Main block diagram of an adaptive filter

In this study, 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 proposed algorithms.

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 $\bar{e}_{L,n}$ of this system at time n is:

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

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

$$d_n = v_n + w_{opt,L}^T x_{L,n} \quad (2)$$

where, $w_{opt,L} = [w_{opt,1}, w_{opt,2}, \dots, w_{opt,L}]^T$ 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 $g_{L,n}$ and according to:

$$w_{L,n} = w_{L,n-1} + g_{L,n} \bar{e}_{L,n} \quad (3)$$

The different algorithms are distinguished by the gain calculation.

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:

$$g_{L,n} = \frac{\mu}{L \pi_{x,n} + c_0} x_{L,n} \quad (4)$$

Where:

μ = Referred to as the adaptation step

c_0 = 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]:

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

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

The NS-FRLS algorithm: The filter $w_{L,n}$ is calculated by minimizing the weighted least squares criterion according to^[11]:

$$J_n(w) = \sum_{i=1}^n \lambda^{n-i} (d_i - w_{L,n}^T x_{L,i})^2 \quad (6)$$

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

$$g_{L,n} = \underbrace{R_{L,n}^{-1} x_{L,n}}_{RLS} = \underbrace{\gamma_{L,n} \tilde{k}_{L,n}}_{FRLS} \quad (7)$$

where, $R_{L,n}$ is an estimate of the correlation matrix of the input signal vector. The variables $\gamma_{L,n}$ and $\tilde{k}_{L,n}$ respectively indicate the likelihood variable and

normalized Kalman gain vector. This latter is calculated, independently of the filtering part $w_{L,n}$, by a FRLS algorithm using forward/backward linear prediction analysis over the signal x_n ^[11]. 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. This variable is given by:

$$\xi_n = \bar{e}_{L,n} - \bar{e}_{L,n}^f \begin{cases} = 0 & \text{theory} \\ \neq 0 & \text{practical} \end{cases} \quad (9)$$

with $\bar{e}_{L,n}^f = [(1-\mu_s)\bar{e}_{L,n}^f + \mu_s\bar{e}_{L,n}^f]$ and $0 \leq \mu_s \leq 1$. It is obtained by using some redundant formulae of the FRLS algorithms. We can calculate differently the backward a priori prediction errors in tree ways ($\bar{e}_{L,n}$, $\bar{e}_{L,n}^f$ and $\bar{e}_{L,n}^i$). 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 $b_{L,n}$. 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]:

$$\lambda = 1 - 1/2L \quad (10)$$

The resulting stabilized FRLS (NS-FRLS) algorithms have a complexity of 8L; it is shown in Algorithm 1. Note that numerical stabilization of the algorithm limits the range of the forgetting factor λ (10) and consequently their convergence speed and tracking ability.

Proposed algorithms:

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.

Algorithm 1: NS-FRLS algorithm (8L)

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

$$w_{L,0} = a_{L,0} = b_{L,0} = \tilde{k}_{L,0} = 0_{L \times 1}$$

Variables available at the discrete-time index n:

$$a_{L,n-1}; b_{L,n-1}; \tilde{k}_{L,n-1}; \gamma_{L,n-1}; \alpha_{L,n-1}; \beta_{L,n-1}; w_{L,n-1}$$

New information: x_n, d_n .

Prediction part:

Modeling of x_n, x_{n-L}

$$\bar{e}_{L,n} = x_n - a_{L,n-1}^T x_{L,n-1};$$

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

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

$$\bar{e}_{L,n} = x_{n-L} - b_{L,n-1}^T x_{L,n};$$

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

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

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

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

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

Filtering part:

$$\bar{e}_{L,n} = d_n - w_{L,n-1}^T x_{L,n}; w_{L,n} = w_{L,n-1} + \bar{e}_{L,n} \gamma_{L,n} \tilde{k}_{L,n}^+$$

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:

$$\begin{bmatrix} \tilde{k}_{L,n} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{k}_{L,n-1} \end{bmatrix} + \frac{\bar{e}_{L,n}}{\lambda \alpha_{L,n-1}} \begin{bmatrix} 1 \\ -a_{L,n-1} \end{bmatrix} - \frac{\bar{e}_{L,n}}{\lambda \beta_{L,n-1}} \begin{bmatrix} -b_{L,n-1} \\ 1 \end{bmatrix} \quad (11)$$

and if we discard all backward prediction variables from (11) and use only the forward variables to compute the normalized Kalman gain:

$$\begin{bmatrix} \tilde{k}_{L,n} \\ * \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{k}_{L,n-1} \end{bmatrix} + \frac{\bar{e}_{L,n}}{\lambda \alpha_{L,n-1}} \begin{bmatrix} 1 \\ -a_{L,n-1} \end{bmatrix} \quad (12)$$

This algorithm is not very robust with nonstationarity input signal like speech signals. The first difficulty comes from $\alpha_{L,n} = \lambda \alpha_{L,n-1} \rightarrow 0$. 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:

$$\frac{\bar{e}_{L,n}}{\lambda\alpha_{L,n-1}} \rightarrow \frac{\bar{e}_{L,n}}{\lambda\alpha_{L,n-1} + c_a} \quad (13)$$

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 $a_{L,n}$ return back to zero by doing the following operation:

$$a_{L,n} \rightarrow \eta a_{L,n} \quad (14)$$

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

$$\gamma_n = \frac{1}{1 + \tilde{k}_n^T x_n} \quad (15)$$

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

$$\begin{bmatrix} \tilde{k}_{L,n} \\ c_{L,n} \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{k}_{L,n-1} \end{bmatrix} + \frac{\bar{e}_{L,n}}{\lambda\alpha_{L,n-1} + c_a} \begin{bmatrix} 1 \\ -a_{L,n-1} \end{bmatrix} \quad (16)$$

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:

$$x_{L+1,n} = x \begin{bmatrix} x_{L,n}^T & x_{n-L} \end{bmatrix}^T \quad (17a)$$

$$x_{L+1,n} = \begin{bmatrix} x_n & x_{L,n-1}^T \end{bmatrix}^T \quad (17b)$$

By multiplying on the left, the members of left and right of the expression (16) by Eq. 17a and b respectively, the following equality is obtained:

$$x_{L,n}^T \tilde{k}_{L,n} + c_{L,n} x_{n-L} = x_{L,n-1}^T \tilde{k}_{L,n-1} + \frac{\bar{e}_{L,n}^2}{\lambda\alpha_{L,n-1} + c_a} \quad (18)$$

Algorithm 2: M-SMFTF algorithm (6L)

Initialization:

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

Variables available at the discrete-time index n :

$$a_{L,n-1}; \tilde{k}_{L,n-1}; \gamma_{L,n-1}; \alpha_{L,n-1}; w_{L,n-1}$$

New information: x_n, d_n .

Prediction part:

$$\bar{e}_{L,n} = x_n - a_{L,n-1}^T x_{L,n-1};$$

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

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

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

Filtering part:

$$\bar{e}_{L,n} = d_n - w_{L,n-1}^T x_{L,n}; w_{L,n} = w_{L,n-1} + \bar{e}_{L,n} \gamma_{L,n} \tilde{k}_{L,n}$$

By manipulating the relation (18), we obtain a new recursive formula for calculating the likelihood variable as given below:

$$\gamma_{L,n} = \frac{\gamma_{L,n-1}}{1 + \delta_{L,n} \gamma_{L,n-1}} \quad (19)$$

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

The computational complexity of the M-SMFTF algorithm is 6L; it is shown in Algorithm 2.

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

$$a_{L,n} = \begin{bmatrix} a_{P,n} \\ 0_{L-P} \end{bmatrix} \quad (21a)$$

$$\begin{bmatrix} \tilde{k}_{L,n} \\ c_{L,n} \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{k}_{L,n-1} \end{bmatrix} + \frac{\bar{e}_{P,n}}{\lambda\alpha_{P,n-1} + c_a} \begin{bmatrix} 1 \\ -a_{P,n-1} \\ 0_{L-P} \end{bmatrix} \quad (21b)$$

where, P is much smaller than L. The first (P+1) components of the $\tilde{k}_{L,n}$ are updated using the reduced size forward variables, the last components are just a shifted version of the (P+1)th component of $\tilde{k}_{L,n}$. 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 $\tilde{k}_{L,n}$. The second likelihood variable, $\gamma_{L,n}$, is used to update the forward predictor $a_{p,n}$ of order P and the transversal filter $w_{L,n}$. The computational complexity of this algorithm is $(2L+4P)$; it is shown in Algorithm 3.

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:

$$\Delta \phi_n = F(n) \Delta \phi_{n-1} + e(n) \quad (22)$$

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

$$\Delta \phi_n = \begin{bmatrix} \Delta \phi_n^a \\ \Delta \phi_n^k \end{bmatrix}^T \quad (23)$$

Where:

$$\Delta \phi_n^a = \begin{bmatrix} \Delta a_{L,n} \\ \Delta \alpha_{L,n} \end{bmatrix} \quad (24a)$$

Algorithm 3: RM-SMFTF algorithm $(2L+4P; P < L)$

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

$$\gamma_{p,0} = 1; \alpha_{p,0} = \lambda^p E_0; \gamma_{L,0} = 1; w_{L,0} = \tilde{k}_{L,0} = 0_L; a_{p,0} = 0_P.$$

Variables available at the discrete-time index n :

$$a_{L,n-1}; \tilde{k}_{L,n-1}; \gamma_{L,n-1}; \alpha_{L,n-1}; w_{L,n-1};$$

New information: x_n, d_n .

Prediction part:

$$\bar{e}_{p,n} = x_n - a_{p,n-1}^T X_{p,n-1};$$

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

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

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

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

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

Filtering part:

$$\bar{e}_{L,n} = d_n - w_{L,n-1}^T X_{L,n}; w_{L,n} = w_{L,n-1} + \bar{e}_{L,n} \gamma_{L,n} \tilde{k}_{L,n}$$

$$\Delta \phi_n^k = \begin{bmatrix} \Delta \tilde{k}_{L,n} \\ \Delta \gamma_{L,n} \end{bmatrix} \quad (24b)$$

represent respectively the errors cumulated up until the time n in the forward and Kalman recursive variables. The $(2L+2) \times (2L+2)$ dimensional matrix $F(n)$ given by:

$$F(n) = \begin{bmatrix} F_{11}(n) & F_{12}(n) \\ F_{21}(n) & F_{22}(n) \end{bmatrix} \quad (25)$$

represents the transition matrix. The system (22) is said to be stable, in the mean sense, if the eigenvalues of $E\{F(n)\}$, in the steady state, are all less than one in magnitude^[9]. The operator $E\{\cdot\}$ 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 $(\Delta a_{L,n}, \Delta \alpha_{L,n})$ and the Kalman variables $(\Delta \tilde{k}_{L,n-1}, \Delta \gamma_{L,n-1})$ by the following linear first order models deduced from differentiating $(a_{L,n}, \alpha_{L,n})$ and $(\tilde{k}_{L,n}, \gamma_{L,n})$ respectively:

$$\Delta a_{L,n} = \eta (I_L - \gamma_{L,n-1} \tilde{k}_{L,n-1} x_{L,n-1}^T) \Delta a_{L,n-1} + p_a(n) \quad (26)$$

$$\Delta \alpha_{L,n} = \lambda \Delta \alpha_{L,n-1} + p_\alpha(n) \quad (27)$$

$$\Delta \tilde{k}_{L,n} = M^k \Delta \tilde{k}_{L,n-1} + p_k(n) \quad (28)$$

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

Where:

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

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

By assuming that the perturbation terms $(p_a(n), p_\alpha(n), p_k(n))$ and $p_\gamma(n)$ remain limited.

In asymptotic mode, we can write:

$$\eta (I_L - \gamma_{L,n-1} \tilde{k}_{L,n-1} x_{L,n-1}^T) \rightarrow \eta \lambda I_L \quad (32)$$

$$E\{c^\gamma(n)\} \rightarrow \frac{1}{\lambda^{-1} + \lambda - 1} \quad (33)$$

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:

$$\lambda > 1 - \frac{1 + \sqrt{1 + \left(\frac{1}{\eta^2} - 1\right)(L+2)}}{(L+2)} \quad (34)$$

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, 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.

MATERIALS AND METHODS

To confirm the validity of our analysis and demonstrate the improved numerical performance, some simulations are carried out. All plots show the mean squared modeling 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 (34) with the stationary input.

RESULTS

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 ($\mu = 1$) and NS-FRLS ($\lambda = 1-1/3L$) algorithms are tuned to obtain fastest convergence. The nonstationarity of the system to be modeled 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 ($\lambda = 1-1/P$). Figure 2 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.

In this simulation, we compare the convergence performance of the NLMS algorithm and RM-SMFTF algorithm with different values for the leakage η . Figure 3 presents the results obtained with the stationary USASI noise 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 $P = 40$, 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.

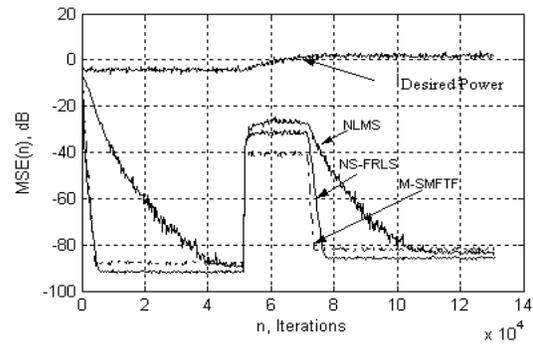


Fig. 2: Comparative performance of the algorithms for USASI noise, $L = 256$. M-SMFTF: $\lambda = 0.9961$, $\eta = 0.985$, $c_a = 0.5$, $E_0 = 1$; NS-FRLS: $\lambda = 0.9987$; NLMS: $\mu = 1$

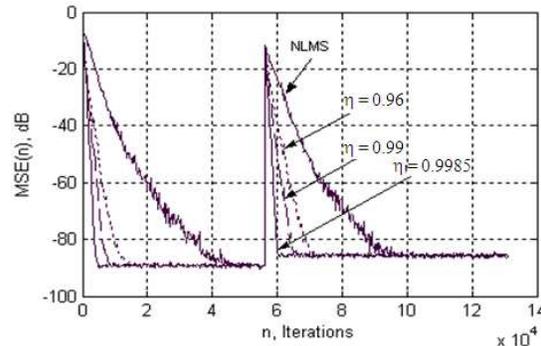


Fig. 3: Comparative performance of the RM-SMFTF and NLMS for USASI noise, $L = 256$. NLMS: $\mu = 1$; RM-SMFTF: $P = 40$, $\lambda = 0.975$, $c_a = 0.5$, $E_0 = 0.2$, with different values for η

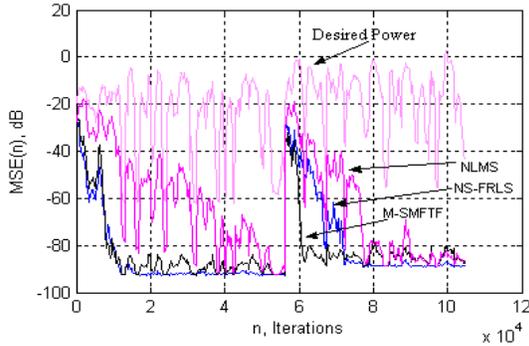


Fig. 4: Comparative performance of the algorithms with speech input, $L = 256$. NLMS: $\mu = 1$; M-SMFTF: $\lambda = 0.9961$, $\eta = 0.96$, $c_a = 0.1$, $E_0 = 0.5$; NS-FRLS: $\lambda = 0.9996$

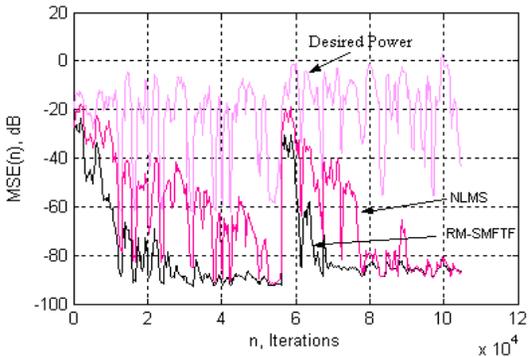


Fig. 5: Comparative performance of the RM-SMFTF and NLMS with speech input, $L = 256$. NLMS: $\mu = 1$; RM-SMFTF: $P = 20$, $\lambda = 0.950$, $\eta = 0.99$, $c_a = 0.1$, $E_0 = 1$

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. 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 and RM-SMFTF algorithms are $\lambda = 1-1/L$ and $\lambda = 1-1/P$ respectively, but the leakage η and the constant c_a must be carefully chosen.

In Fig. 4, 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. In Fig. 5, we observe that the re-convergence of RM-SMFTF is again faster than NLMS.

DISCUSSION

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 $P \ll L$. 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 . 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.

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 ($P \ll L$) 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 render it very interesting for applications such as acoustic echo cancellation. The simulation has shown that the performances of proposed algorithm are better than those of the normalized least mean square algorithm.

REFERENCES

1. Haykin, S., 2001. Adaptive Filter Theory. 4th Edn., Prentice-Hall, New Jersey, ISBN: 10: 0130901261, pp: 936.
2. Sayed, A.H., 2003. Fundamentals of Adaptive Filtering. John Wiley and Sons, New York, USA., ISBN: 978-0-471-46126-5, pp: 1168.
3. Macchi, O., 1995. The Least Mean Squares Approach with Applications in Transmission. Wiley, New York, ISBN: 10: 0471934038, pp: 476.
4. Treichler, J.R., C.R. Johnson and M.G. Larimore, 2001. Theory and Design of Adaptive Filter. Prentice Hall, New Jersey, ISBN: 10: 0130402656, pp: 475.
5. Ljung, L., M. Morf and D. Falconner, 1978. Fast calculation of gain matrices for recursive estimation schemes. Int. J. Control, 27: 1-19. DOI: 10.1080/00207177808922343

6. Caryannis, G., D. Manolakis and N. Kalouptsidis, 1983. A fast sequential algorithm for least squares filtering and prediction. *IEEE Trans. Acoust. Speech Signal Process.*, 31: 1394-1402. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1164224
7. Cioffi, J. and T. Kailath, 1984. Fast RLS Transversal Filters for adaptive filtering. *IEEE Trans. Acoust. Speech Signal Process.*, 32: 304-337. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1164334
8. Benallal, A. and A. Gilloire, 1988. A New method to stabilize fast RLS algorithms based on a first-order model of the propagation of numerical errors. *Proceeding of the International Conference on Acoustics, Speech and Signal Processing*, Apr. 11-14, IEEE Xplore Press, USA., pp: 1373-1376. DOI: 10.1109/ICASSP.1988.196852
9. Slock, D.T.M. and T. Kailath, 1991. Numerically stable fast transversal filters for recursive least squares adaptive filtering. *IEEE Trans. Signal Process.*, 39: 92-114. DOI: 10.1109/78.80769
10. Liavas, A.P. and P.A. Regalia, 1998. Numerical stability issues of the conventional recursive least squares algorithm. *Proceeding of the International Conference on Acoustics, Speech and Signal Processing*, May 12-15, IEEE Xplore Press, Seattle, WA., pp: 1409-1412. DOI: 10.1109/ICASSP.1998.681711
11. Arezki, M., A. Benallal, P. Meyrueis, A. Guessoum and D. Berkani, 2007. Error propagation analysis of fast recursive least squares algorithms. *Proceeding of the 9th IASTED International Conference on Signal and Image Processing*, Aug. 20-22, Honolulu, Hawaii, USA., pp: 97-101. <https://actapress.com/Abstract.aspx?paperId=31050>
12. Binde, S., 1995. A numerically stable fast transversal filter with leakage correction. *IEEE Signal Process. Lett.*, 2: 114-116. DOI: 10.1109/97.388912
13. Moustakides, G.V. and S. Theodoridis, 1991. Fast Newton transversal filters-A new class of adaptive estimation algorithms. *IEEE Trans. Signal Process.*, 39: 2184-2193. DOI: 10.1109/78.91175
14. Mavridis, P.P. and G.V. Moustakides, 1996. Simplified Newton-type adaptive estimation algorithms. *IEEE Trans. Signal Process.*, 44: 1932-1940. DOI: 10.1109/78.533714
15. Benallal, A. and A. Benkrir, 2007. A simplified FTF-type algorithm for adaptive filtering. *Signal Process.*, 87: 904-917. <http://portal.acm.org/citation.cfm?id=1225027>
16. Slock, D.T.M., 1993. On the convergence behavior of the LMS and the NLMS algorithms. *IEEE Trans. Signal Process.*, 42: 2811-2825. DOI: 10.1109/78.236504
17. Gilloire, A., E. Moulines, D. Slock and P. Duhamel, 1996. State of Art in Echo Cancellation. In: *Digital Signal Processing in Telecommunication*, Figuers-Vidal, A.R. (Ed.). Springer, Berlin, ISBN: 10: 3540760377, pp: 45-91.