

A SIMPLIFIED FTF-TYPE ALGORITHM FOR ADAPTIVE FILTERING

Ahmed Benallal, Abdelhak Benkrid

College of Technology Dammam, Saudi Arabia

phone: + (966)568400517, fax: + (966)38681433, email: (a_benallal, benkridah@hotmail.com)

ABSTRACT

Fast RLS algorithms are known to present numerical instability and this instability is originated in the forward prediction parameters. In this paper, A simplified FTF-Type algorithm for adaptive filtering is presented. The basic idea behind the proposed algorithm is to avoid using the backward variables. The algorithm obtained is less complex than the existing numerically stable fast FTF and shows the same performances.

1. INTRODUCTION

Fast recursive least squares (FRLS) adaptive filtering algorithms represent an attractive way to compute the least squares solution efficiently. Conventional RLS requires $O(N^2)$ computations per sample, its fast versions require only $O(N)$ operations. Examples of such fast algorithms include the fast a posteriori error sequential technique (FAEST) [3] and the fast transversal filter algorithm (FTF) [4]. The low complexity that is achieved by these algorithms is a direct consequence of the shift invariance structure of the input signal vector used in FIR filtering implementations. The least squares solution is obtained by updating the Kalman gain vector, which is used to update the transversal filter, by using forward and backward prediction vectors.

In this contribution, we propose an algorithm 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. Our main goal is to obtain a robust FTF-like adaptive algorithm when used with speech signal to solve acoustic echo cancellation problems. By using only forward prediction variables and adding a small regularisation constant and a leakage factor, we obtain a robust numerically stable adaptive algorithm that shows the same performances as FRLS algorithms.

The computational complexity of the proposed algorithm is $7N$ when used with a full size predictor. The proposed algorithm is less complex than the $8N$ numerically stable FTF (SFTF) algorithms [5,6] and its complexity can be significantly reduced to $2N+5P$ when used with a reduced P-size forward predictor.

2. THE FAST RLS ALGORITHMS

Figure 1 shows the basic schematic diagram of an adaptive filter, where $x(n)$, $d(n)$, and $\varepsilon_N(n)$ are, respectively, the input, desired output and error signals of the adaptive Filter for time instant n . In most adaptive identification applications, the adaptive algorithm enables the transversal filter to be estimated by the N -size vector $H_N(n)$ using a criterion based on the a priori estimation error $\varepsilon_N(n)$. This estimation error is written, for each sample n :

$$\varepsilon_N(n) = d(n) - H_N^T(n-1)X_N(n) \quad (1)$$

where $X_N^T(n) = [x(n), x(n-1), \dots, x(n-N+1)]$ represents the N last samples of the input signal. The filter is updated at each instant by feedback of the estimation error proportional to the adaptation gain, denoted as $C_N(n)$, and according to:

$$H_N(n) = H_N(n-1) - C_N(n)\varepsilon_N(n) \quad (2)$$

Different algorithms are distinguished by the gain calculation $C_N(n)$ as described in the following paragraphs.

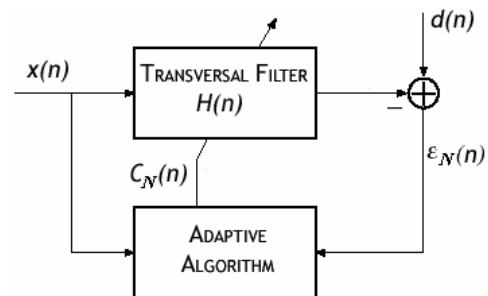


Figure 1: Adaptive Filtering

For the NLMS algorithm, the adaptation gain is given by:

$$C_N(n) = -\mu \frac{X(n)}{X^T(n)X(n) + c} \quad (3)$$

where μ is referred to as the adaptation step and c is a small positive constant used to avoid division by 0 in absence of the input signal.

Recursive least squares (RLS) algorithms are based on a minimization of the criterion of the least squares with an exponential forgetting factor λ ($1 < \lambda \leq 1$):

$$J(n) = \sum_{i=1}^n \lambda^{n-i} [d(i) - H_N^T(n)X_N(i)]^2 \quad (4)$$

In the conventional RLS, the adaptation gain for (2) is given by:

$$C(n) = \gamma_N(n) \tilde{C}_N(n) \quad (5)$$

where $\tilde{C}_N(n)$ and $\gamma(n)$, called the dual Kalman gain and the likelihood variable respectively, are updated by propagating an $N \times N$ inverse covariance matrix. This solution requires $O(N^2)$ operations per iteration which is high computational complexity. However, with the introduction of *fast* versions of the algorithm, the least squares solution remains an attractive solution for major practical applications.

The main advantage of the RLS algorithm is its very fast convergence which is independent of the input signal nature. FRLS algorithms are derived from the RLS algorithm by the introduction of forward and backward predictors in the calculation of the adaptation gain $C_N(n)$; the likelihood variable and the dual Kalman gain are computed, independently of the filtering part $H_N(n)$, by a FRLS algorithm using a linear prediction analysis over the input signal $x(n)$:

$$\begin{bmatrix} \tilde{C}_N(n) \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{C}_N(n-1) \end{bmatrix} - \frac{\bar{e}_N(n)}{\lambda \alpha_N(n-1)} \begin{bmatrix} 1 \\ -a_N(n-1) \end{bmatrix} + \frac{\bar{r}_N(n)}{\lambda \beta_N(n-1)} \begin{bmatrix} -b_N(n-1) \\ 1 \end{bmatrix} \quad (6)$$

where $a_N(n)$ and $b_N(n)$ are, respectively, forward and backward predictors, and $\bar{e}_N(n)$ and $\bar{r}_N(n)$ are, respectively, forward and backward prediction errors and $\alpha_N(n)$ and $\beta_N(n)$ designed, respectively, forward and backward prediction error variances.

The forward and backward predictors, updated recursively in the FRLS, are defined by:

$$a_N(n) = R_N^{-1}(n-1) \Gamma_N^f(n), \Gamma_N^f(n) = \sum_{i=1}^n \lambda^{n-i} X_N(i-1)x(i)$$

$$b_N(n) = R_N^{-1}(n) \Gamma_N^b(n), \Gamma_N^b(n) = \sum_{i=1}^n \lambda^{n-i} X_N(i)x(i-N)$$

where $R_N(n)$ is an estimate of the correlation matrix of the input signal vector. $\Gamma_N^f(n)$ is an estimate of the cross-correlation of $x(n)$ with its past values $X_N(n-1)$ and $\Gamma_N^b(n)$ is an estimate of the cross-correlation of $x(n-N)$ with its future values $X_N(n)$. For stationary input signals, backward and forward power variances are equal and the backward and forward predictors have the same coefficients but in inverse order [1,2].

As mentioned before, all original FRLS algorithms are numerically instable. It has been shown [5, 6] that numerical stability can be provided by feeding back the numerical errors in the computation of the prediction variables. The resulting stabilized FRLS algorithms have a complexity of $8N$.

3. SIMPLIFIED FTF-TYPE ALGORITHM

Before giving the new simplified algorithm, we must make two important observations. If we discard completely the prediction parts from the algorithm, this case correspond theoretically to the white noise input signal, the Kalman gain (6) becomes:

$$\begin{bmatrix} \tilde{C}_N(n) \\ * \end{bmatrix} = \begin{bmatrix} -\frac{x(n)}{\lambda \alpha_N(n-1)} \\ \tilde{C}_N(n-1) \end{bmatrix} \quad (7)$$

And after the N first iterations, a shifted version of the first component is propagated to all other components:

$$\begin{bmatrix} \tilde{C}_N(n) \\ * \end{bmatrix} = \begin{bmatrix} \frac{x(n)}{\lambda \alpha_N(n-1)}, \frac{x(n-1)}{\lambda \alpha_N(n-2)} \dots \frac{x(n-N)}{\lambda \alpha_N(n-N-1)} \end{bmatrix}^T \quad (8)$$

In the steady-state, for a stationary input signal, the forward error variance can be approximated by quantity proportional to the signal power:

$$\alpha_N(n) = \lambda \alpha_N(n-1) + \gamma_N(n) x^2(n) \rightarrow \gamma_F \sigma_X^2 / (1-\lambda) \quad (9)$$

where γ_F is the steady-state value of the slowly variable $\gamma_N(n)$. Then the terms $\lambda \alpha_N(n-i)$ converge towards a value proportional to the input signal power and we can approximately write:

$$C_N(n) = \gamma_N(n) \tilde{C}_N(n) \approx -wX(n) / \sigma_X^2 \quad (10)$$

where w is a positive constant.

This adaptation gain has a form similar to the adaptation gain of the NLMS algorithms. This is an important observation which means that even with no forward and backward predictors, the FTF algorithm must work but may be with performances just close to those of an NLMS algorithm.

Now assume we have the adaptation gain updated with the two predictors. We know that physical impulse responses are globally decreasing with the order of the filter. This case is particularly verified in the acoustic echo cancellation problem. In this application, predictor sizes are much smaller than the size of the transversal filter for speech signal. This propriety was used to develop a class of algorithms called Fast Newton transversal filter algorithm [7, 8] where the input signal is modelised by an AR model with 10 to 20 coefficients.

From relation (6), we can see that the most significant components, the last ones, of the backward predictor affect the last terms of the Kalman Gain and this contribution is not transmitted to other lower positions of $\tilde{C}_N(n)$ because of the down shift property in the calculation of the dual Kalman gain. The first components of $\tilde{C}_N(n)$ come from the forward prediction error signal $-\bar{e}_N(n) / \lambda \alpha_N(n-1)$ and the most significant components of the forward predictor. This means that the most significant components of the adaptive filter $H_N(n)$ are updated using forward prediction variables, the backward prediction variables affect the

last components of $H_N(n)$ which in most practical applications have very small values.

This observation suggests discarding completely the backward predictor from the FTF algorithm. Discarding the backward predictor does not mean that the last components $H_N(n)$ are not updated, but they are updated by components coming from lower positions of $\tilde{C}_N(n)$.

In the proposed algorithm, we discard all backward prediction variables from (6) and use only the forward variables to compute the dual Kalman gain :

$$\begin{bmatrix} \tilde{C}_N(n) \\ * \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{C}_N(n-1) \end{bmatrix} - \frac{\bar{e}_N(n)}{\lambda \alpha_N(n-1)} \begin{bmatrix} 1 \\ -a_N(n-1) \end{bmatrix} \quad (11)$$

where

$$\alpha_N(n) = \lambda \alpha_N(n-1) + \gamma_N(n-1) \bar{e}_N^2(n)$$

$$a_N(n) = a_N(n-1) - \bar{e}_N(n) \gamma_N(n-1) \tilde{C}_N(n-1)$$

$$\gamma_N(n) = \frac{1}{1 - \tilde{C}_N^T(n) X_N(n)}$$

This algorithm is not very robust with non-stationary input signal like speech signals. Suppose that after some converge, the signal input becomes 0 or very small, this situation can be observed with input speech signals in silent periods:

$$\begin{cases} \bar{e}_N(n) = 0, \tilde{C}_N(n) = 0, \gamma_N(n) = 1, a_N(n) = a_N(n-1) \\ \alpha_N(n) = \lambda \alpha_N(n-1) \rightarrow 0 \end{cases} \quad (12)$$

The first difficulty comes from $\alpha_N(n) = \lambda \alpha_N(n-1) \rightarrow 0$.

This convergence to 0, put FTF algorithms and their numerically stable versions in very difficult situation. Instability may occur since we are trying to perform numerical division by very small values. It is important to avoid divisions by very small values of $\lambda \alpha_N(n-1)$. To guard against this possibility, like in the NLMS algorithm, we append a positive constant to dividing term $\lambda \alpha_N(n-1)$:

$$\frac{\bar{e}_N(n)}{\lambda \alpha_N(n-1)} \rightarrow \frac{\bar{e}_N(n)}{\lambda \alpha_N(n-1) + c} \quad (13)$$

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 0 in order to the algorithm to begin adapting. In these conditions, when the input signal vanishes and reappears after a long period of time, the algorithm may diverge because of these non-zero values of the forward 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 $a_N(n)$ return back to zero by doing the following operation:

$$a_N(n) \rightarrow \eta a_N(n) \quad (14)$$

where η is a close to 1 constant often called Leakage factor [1,9].

The simplified FTF-Type algorithm (SMFTF) is given in table 1 for a full size forward predictor. An error propagation analysis of the prediction part of the SMFTF shows that the algorithm is globally stable in the mean sense and the variance of the numerical errors in the forward predictor, with the assumption of a white Gaussian input signal and a forgetting factor close to 1, is stable under the following condition [10] :

$$\lambda \geq 1 - \frac{1 + \sqrt{1 + (N+2)(\frac{1}{\eta^2} - 1)}}{N+2} \quad (15)$$

Reduced size predictors in the FTF algorithms have been successfully used in the FNTF algorithms [7,8]. The SMFTF algorithm can easily used with reduced size prediction part (table 2). If we note P the order of the predictor and N the size of adaptive filter, the First P components of the dual Kalman gain are updated using the reduced size forward variables, the N-P last components are just a shifted version of the Pth component the dual Kalman gain. For this algorithm, we need two likelihood variables: the first one $\gamma_P(n)$, of order P, is used to update the backward prediction error variance. The second, $\gamma_N(n)$, of order N, is used with the dual Kalman Gain to update the forward predictor and the transversal filter.

The proposed reduced size predictor algorithm is summarized in Table 3. Notice that the total complexity is to $3N+4P$ where P is the size of the forward predictor for version 1 and becomes $2N+5P$ when we use the Pth shifted component of the dual Kalman gain to compute the likelihood variable $\gamma_N(n)$ (version 2).

Table 1 : Simplified FTF-Type algorithm (SMFTF)

The SMFTF Algorithm	
Initialisation: $a_N(0) = 0, \tilde{C}_N(0) = 0, \gamma_N(0) = 1,$ $\alpha_N(0) = E_0 \lambda^N$	
Prediction Part	
$\bar{e}_N(n) = x(n) - a_N^T(n-1) X_N(n-1)$	
$\alpha_N(n) = \lambda \alpha_N(n-1) + \gamma_N(n-1) \bar{e}_N^2(n)$	
$\begin{bmatrix} \tilde{C}_N(n) \\ * \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{C}_N(n-1) \end{bmatrix} - \frac{\bar{e}_N(n)}{\lambda \alpha_N(n-1) + c} \begin{bmatrix} 1 \\ -a_N(n-1) \end{bmatrix}$ (we can divide with $\alpha_N(n) + c$ instead of $\lambda \alpha_N(n-1) + c$)	
$a_N(n) = \eta [a_N(n-1) - \bar{e}_N(n) \gamma_N(n-1) \tilde{C}_N(n-1)]$	
$\gamma_N(n) = \frac{1}{1 - \tilde{C}_N^T(n) X_N(n)}$	
Filtering	
$\varepsilon_N(n) = d(n) - H_N^T(n-1) X_N(n)$	
$H_N(n) = H_N(n-1) - \varepsilon_N(n) \gamma_N(n) \tilde{C}_N(n)$	

<p>Reduced Prediction order SMFTF Algorithm</p> <p>Initialisation:</p> $a_p(0) = 0, \tilde{C}_N(0) = 0, \gamma_p(0) = \gamma_N(0) = 1,$ $\alpha_p(0) = E_0 \lambda^p, H_N(0) = 0$
$X_p(n) \text{ are the first } P \text{ components of } X_N(n)$
$\bar{e}_p(n) = x(n) - a_p^T(n-1)X_p(n-1)$
$\alpha_p(n) = \lambda \alpha_p(n-1) + \gamma_p(n-1) \bar{e}_p^2(n)$
$\begin{bmatrix} \tilde{C}_N(n) \\ * \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{C}_N(n-1) \end{bmatrix} - \frac{\bar{e}_p(n)}{\lambda \alpha_p(n-1) + c} \begin{bmatrix} 1 \\ -a_p(n-1) \\ 0_{N-p} \end{bmatrix}$ <p>(we can divide with $\alpha_p(n) + c$ instead of $\lambda \alpha_p(n-1) + c$)</p>
$a_p(n) = \eta [a_p(n-1) - \bar{e}_p(n) \gamma_N(n-1) \tilde{C}_p(n-1)]$ <p>$\tilde{C}_p(n)$ are the first P components of $\tilde{C}_N(n)$</p>
$\theta_p(n) = 1 - \tilde{C}_p^T(n) X_p(n), \quad \gamma_p(n) = \frac{1}{\theta_p(n)}$
<p>*version1:</p> $\gamma_N(n) = \frac{1}{\theta_p(n) - \sum_{i=p+1}^N \tilde{C}_N^i(n) x(n-i+1)}$
<p>* version 2:</p> $v(n) = \tilde{C}_N^{p+1}(n) x(n-p)$ $\gamma_N(n) = \frac{1}{\theta_p(n) - \sum_{i=1}^{N-p} v(n-i+1)}$
<p>Filtering</p>
$\varepsilon_N(n) = d(n) - H_N^T(n-1) X_N(n)$
$H_N(n) = H_N(n-1) - \varepsilon_N(n) \gamma_N(n) \tilde{C}_N(n)$

4. SIMULATION RESULTS

We used in our simulation two types of input signal : a stationary noise with an average speech spectrum (USASI noise) and a typical speech signal. These two signals, sampled at 16 KHz, are filtered by two impulse responses to obtain the desired signals. The first impulse response, represents a low pass filter of size 32, and the second represents a real impulse response measured in a car and truncated to $N = 256$ samples. The power signal $\sigma_x^2 = 0.33$ for USASI noise. The MSE of the filtering error is plotted against the number of iterations, averaged over an ensemble of 256 samples.

In our simulation, we have compared the proposed SMFTF algorithm with the NLMS and the numerically stable FTF (SFTF) algorithms. For the NLMS we choose the adaptation step that corresponds to the fastest convergence $\mu = 1$. The forgetting factor for the SFTF algorithm is chosen to ensure numerical stability by using $\lambda = 1 - 1/3N$ with USASI

noise and $\lambda = 1 - 1/10N$ with speech signal. The forgetting factor and the leakage factor for the SMFTF algorithm are chosen according to (15) with the stationary USASI noise. The choice of the forgetting factor is less restrictive than for SFTF algorithm, and we choice $\lambda = 1 - 1/N$ for speech signal. The constant E_0 is fixed to 1 in all our simulations for the SFTF and full size predictor SMFTF algorithms.

Figures 1 and 2 present the results obtained with USASI noise for $N=32$ and $N=256$, respectively. These results show that the proposed algorithm achieves better performances than the NLMS algorithm and does not introduced any noticeable performance degradation compared to the SFTF algorithm.

To compare the convergence speed, we simulated a jump by multiplying the desired signal by 1.5 in the steady-state. Figure 3 shows that better performances in convergence speed are obtained for the proposed algorithm. This improvement is due to the use of a smaller forgetting factor 0.996 for the SMFT algorithm. This value is not possible to use with the SFTF algorithm due to stability problems.

With the speech signal, the leakage factor and the regularisation constant must be carefully chosen. In our simulations, we have noticed that the leakage factor must be chosen smaller enough than the forgetting factor to assure stability of the full size SMFTF algorithm. The regularisation constant c is used to limit the dynamic of adaptation gain: great values give a more robust algorithm to the non-stationary of the signal, small values of c allow more dynamic of the adaptation gain and may improve the convergence speed. The signal power is a good indicator how to choose the constant c . Figure 4 shows clearly better performances for the proposed algorithm after the jump for the speech input signal.

Figure 5 and 6 summarize the results obtained for the reduced size predictor SMFTF algorithm. These results show that there is not degradation in the final steady-state MSE of the reduced size predictor algorithm even for $P \ll N$. Almost identical performances with the full size predictor algorithm are reached when the forgetting is computed by $1 - 1/P$ and the leakage factor is greater enough than λ . The initial constant E_0 must chosen large enough to avoid initial convergence degradation for the reduced size predictor SMFTF when $P \ll N$.

5. CONCLUSION

We have presented a simplified FTF-type algorithm 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 FTF algorithm, are completely discarded. The proposed algorithm compares favorably with the FTF and NLMS algorithms. It does not present numerical instability problems due to finite precision as is the case for the FTF and FNTF algorithms and is more robust with

speech signal due to the use of a leakage factor and a regularisation constant

REFERENCES

[1] S. Haykin, *Adaptive Filter Theory*, 4th ed. Prentice Hall, 2002.
 [2] M. Belanger, *Adaptive Digital Filters and Signal Analysis*, Marcel Dekker, 1987.
 [3] G. Caryannis, D. Manolakis, and N. Kalouptsidis, "A fast sequential algorithm for least squares filtering and prediction" *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 1394-1402, Dec. 1983.
 [4] J. Cioffi and T. Kailath, "Fast recursive least squares transversal filters for adaptive filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 304-337, Apr. 1984.
 [5] A. Benallal, A. Gilloire "A New method to stabilize fast RLS algorithms based on a first-order model of the propagation of numerical errors" Proc. ICASSP88, New York 1988.
 [6] D.T.M. Stock and T. Kailath, "Numerically stable fast transversal filters for recursive least squares adaptive filtering", *IEEE transactions on signal processing*, vol-39, N°1, January 1991
 [7] T. Petition, A. Gilloire, and S. Theodoridis, "The fast Newton transversal filter: An efficient scheme for acoustic echo cancellation in mobile radio," *IEEE Trans. Signal Processing*, vol. 42, pp. 509-518, Mar. 1993.
 [8] P. P. Mavridis, G. V. Moustakides, "Simplified Newton-Type Adaptive Estimation Algorithms" *IEEE Trans. Signal Processing*, vol. 44, pp. no.8, Aug. 1996.
 [9] S. Blande "A numerically stable FTF with leakage correction" *IEEE Processing Letters*, vol. 2, no. 6, June 1995.
 [10] A. Benalla, A. Benkrid "A Simplified FTF-Type Algorithm For Adaptive Filtering" submitted to Elsevier Signal Processing.

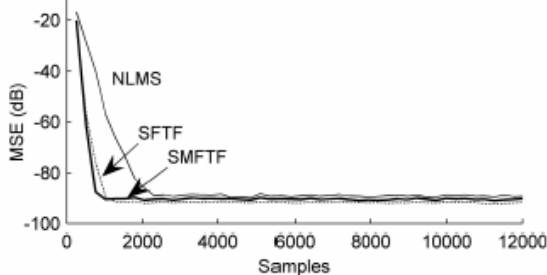


Figure 1 : $x(n)=USASI, N=32$, SFTF: $\lambda = 0.992$, SMFTF: $\lambda = 0.98, \eta = 0.98$ and $c = 0.1$

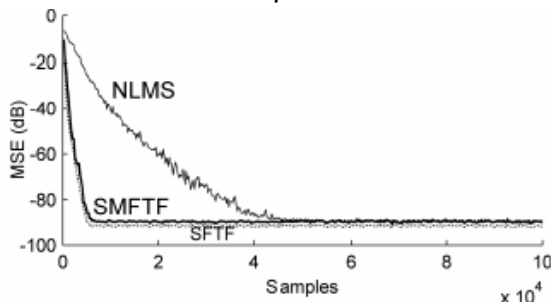


Figure 2: $x(n)=USASI, N=256$, SFTF: $\lambda = 0.9987$, SMFTF: $\lambda = 0.9975, \eta = 0.995$ and $c = 0.1$

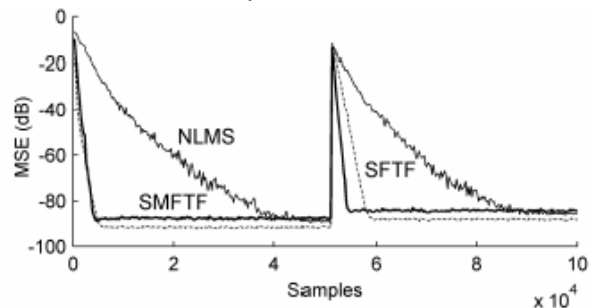


Figure 3 : $x(n)=USASI, N=256$, SFTF: $\lambda = 0.9987$, SMFTF: $\lambda = 0.996, \eta = 0.985$ and $c = 0.5$.

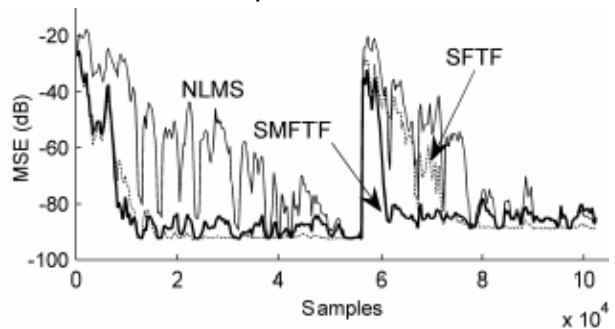


Figure 4: $x(n)=speech, N=256$. SFTF: $\lambda = 0.9996$, SMFTF: $\lambda = 0.996, \eta = 0.96$ and $c = 0.1$.

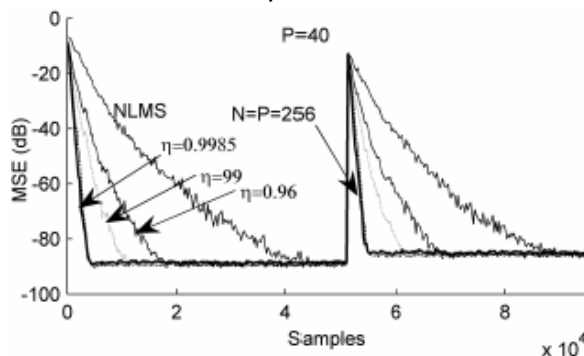


Figure 5. Simulation for the reduced size predictor SMFTF with different values for the leakage factor. $N=256, P=40$. $x(n) = USASI$. SMFTF: $\lambda = 0.975, E_0 = 5$ and $c = 1$.

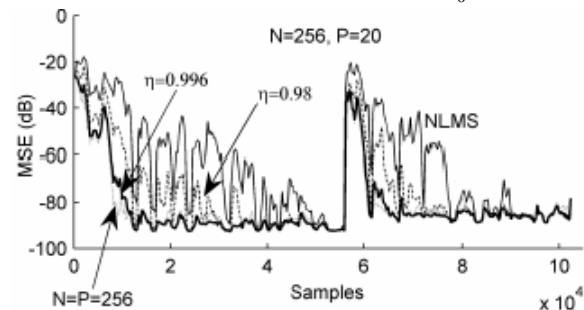


Figure 6. Simulation for the reduced size predictor SMFTF with different values for the leakage factor. $N=256, P=20$. $x(n) = Speech$. SMFTF: $\lambda = 0.95, E_0 = 5$ and $c = 0.1$.