International Conference on
Information, Communications and Signal Processing
ICICS '97
Singapore, 9-12 September 1997

2E3.3

Computational Complexity Reduction of Predictor Based Least Squares Algorithm and Its Numerical Property

Youhua Wang Kazushi Ikeda Kenji Nakayama

Dept. of Electrical and Computer Engineering, Kanazawa University
2-40-20 Kodatsuno, Kanazawa 920, Ishikawa, JAPAN

Abstract—The backward predictor based least squares (BPLS) algorithm, which is derived from the fast recursive least squares (FRLS) algorithms, demonstrates a very stable and robust numerical performance compared with the RLS and FRLS algorithms. However, the computational load of the BPLS algorithms is $O(N^2)$. This makes it difficult to be implemented in real time applications even using today's DSP technology. In order to overcome this difficulty, a method for reducing the computational complexity of the BPLS algorithms is proposed. The result (we call it the fast BPLS algorithm) is consistent with the fast Newton transversal filters (FNTF) algorithms, but the derivation is much simpler to understand. The most important characteristic of the fast BPLS algorithm is its good numerical property. Theoretical analysis and computer simulations show that the fast BPLS algorithm provides a much improved numerical performance compared with the FNTF algorithms under a finite-precision implementation.

1 Introduction

In solving the least squares problem for transversal adaptive filters, the RLS and the FRLS algorithms are well known. However, little attention has been paid to the use of the order-update of the FRLS algorithm. This use leads to the algorithm we called the predictor based least squares (PLS) that consists of the forward PLS (FPLS) and backward PLS (BPLS) algorithms. A comparative study on the numerical performances of the BPLS and the RLS algorithms has been done in [1]. It was shown that three main instability sources encountered in both the RLS and the FRLS algorithms, including the unstable behavior of the conversion factor, the loss of symmetry and the loss of positive definiteness of the inverse correlation matrix, do not exist in the BPLS algorithm. This leads to a much more numerically stable and robust performance of the BPLS algorithm than that of the RLS algorithm.

Unfortunately, the computational load of the PLS algorithm is $O(N^2)$, N is the order of the adaptive filter. This makes it difficult to be implemented in real time applications even using today's DSP technology. In order to overcome this difficulty, a computational complexity reduction of the BPLS algorithm, we call it the fast BPLS algorithm, is proposed in this paper. The assumption for the proposed algorithm is the same as that of the FNTF [2], that is, if the input signal can be sufficiently modeled by an autoregressive of order M, where M is possible to be selected much smaller than the order N of the adaptive filter, then the gain vector can be extended from M to N based on the predictor and the gain vector of order M without sacrificing the performance. However, the derivation presented in this paper is different from that of [2]. Instead of using the Max-Min and Min-Max principle that is somewhat difficult to understand, the derivation shown here is direct and much easy to understand. Furthermore, the derivation of the FNTF algorithms is based on the FRLS algorithms and supposed to be combined with the FRLS algorithms. This will inherently accompany with the instability problem. On the other hand, the proposed algorithm has a much improved numerical performance.

This paper is arranged as follows: In Sec.2, we give the derivation of the fast BPLS algorithm. The numerical property of the fast BPLS algorithm is analyzed in Sec.3. The effects of three main instability sources are considered under a finite precision arithmetic. Sec.4 presents some simulation results of the fast BPLS algorithm using a variety of word-length arithmetic. The comparison on the numerical performances between the fast BPLS and the FNTF algorithms is also addressed.

2 Derivation of Fast BPLS Algorithm

The derivation is based on the BPLS algorithm that is written as

$$\psi_m(n) = \mathbf{c}_m^T(n-1)\mathbf{u}_m(n) + u(n-m) \quad (1)$$

$$B_m(n) = \lambda B_m(n-1) + \gamma_m(n)\psi_m^2(n) \quad (2)$$

$$\gamma_{m+1}(n) = \frac{\lambda B_m(n-1)}{B_m(n)}\gamma_m(n) \quad (3)$$

$$\mathbf{c}_m(n) = \mathbf{c}_m(n-1) - \gamma_m(n)\psi_m(n)\bar{\mathbf{k}}_m(n) \quad (4)$$

$$\bar{\mathbf{k}}_{m+1}(n) = \begin{bmatrix} \bar{\mathbf{k}}_m(n) \\ 0 \end{bmatrix}$$

$$+\frac{\psi_m(n)}{\lambda B_m(n-1)}\begin{bmatrix} c_m(n-1) \\ 1 \end{bmatrix} \quad (5)$$

where $\psi_m(n)$ is the backward a priori prediction error, $B_m(n)$ is the minimum power of $\psi_m(n)$, $\mathbf{c}_m(n)$ is the tap-weight vector of the backward predictor, $\gamma_m(n)$ is the conversion factor, $\bar{\mathbf{k}}_m(n)$ is the normalized gain vector and $\mathbf{u}_m(n)$ is the input vector.

The initial conditions for the BPLS algorithm are as follows: At time $n = 0$, set $\mathbf{c}_m(0) = \mathbf{0}_m$, $B_m(0) = \delta$, $\bar{\mathbf{k}}_m(0) = \mathbf{0}_m$ and $\gamma_m(0) = 1$, where $m = 1, 2, \cdots M$. At each iteration $n \geq 1$, generate the first-order variables as follows:

$$\bar{\mathbf{k}}_1(n) = \frac{u(n)}{\lambda\Phi_1(n-1)} \quad (6)$$

$$\gamma_1(n) = \frac{\lambda\Phi_1(n-1)}{\Phi_1(n)} \quad (7)$$

where $\Phi_1(n)$ is the first-order of the correlation matrix that satisfies

$$\Phi_1(n) = \lambda\Phi_1(n-1) + u^2(n) \quad (8)$$

where $\Phi_1(0) = \delta$.

The use of the normalized gain vector $\bar{\mathbf{k}}_m(n)$ instead of the gain vector $\mathbf{k}_m(n)$ will be explained latter.

Assume that the input signal can be modeled by an AR(M), implying that the use of the predictor of order M is sufficient. The problem is how to extend the gain vector from $\bar{\mathbf{k}}_M(n)$ to $\bar{\mathbf{k}}_N(n)$ based on the knowledge of the M-th order backward predictor with least increase of computation. For $m > M$, the optimum choice of this predictor results

$$\frac{\psi_m(n)}{\lambda B_m(n-1)}\begin{bmatrix} \mathbf{c}_m(n-1) \\ 1 \end{bmatrix} = \frac{\psi_M(n-m+M)}{\lambda B_M(n-m+M-1)} \cdot \begin{bmatrix} \mathbf{0}_{m-M} \\ \mathbf{c}_M(n-m+M-1) \\ 1 \end{bmatrix} \quad (9)$$

To prove (9), we first compute the BPLS algorithm to get $\bar{\mathbf{k}}_{M+1}(n)$ and the predictor of order M. Then, we write (5) for $m = M+1$ as

$$\bar{\mathbf{k}}_{M+2}(n) = \begin{bmatrix} \bar{\mathbf{k}}_{M+1}(n) \\ 0 \end{bmatrix} + \frac{\psi_{M+1}(n)}{\lambda B_{M+1}(n-1)}\begin{bmatrix} \mathbf{c}_{M+1}(n-1) \\ 1 \end{bmatrix} \quad (10)$$

From the assumption, the first term of $\mathbf{c}_{M+1}(n-1)$ is zero, that is

$$\mathbf{c}_{M+1}(n-1) = \begin{bmatrix} 0 \\ \hat{\mathbf{c}}_M(n-1) \end{bmatrix} \quad (11)$$

We want to determine the tap-weight vector of the backward predictor $\mathbf{c}_{M+1}(n-1)$ so that the prediction error $\psi_{M+1}(n)$ and its error power $B_{M+1}(n)$ can be minimized.

Since

$$\begin{aligned}\psi_{M+1}(n) &= \mathbf{c}^T_{M+1}(n-1)\mathbf{u}_{M+1}(n) + u(n-M-1) \\ &= \hat{\mathbf{c}}^T_M(n-1)\mathbf{u}_M(n-1) + u(n-M-1) \quad (12)\end{aligned}$$

the optimum predictor, which uses $u(n-1), \cdots, u(n-M)$ to predict $u(n-M-1)$, is $\mathbf{c}_M(n-2)$ that satisfies

$$\psi_M(n-1) = \mathbf{c}^T_M(n-2)\mathbf{u}_M(n-1) + u(n-M-1) \quad (13)$$

$\psi_M(n-1)$ is the minimum prediction error (least squares solution), that is

$$\psi_M(n-1) = \min[\psi_{M+1}(n)]$$

Under the constraint of using $\psi_M(n-1)$, the minimum prediction error power we can get is

$$B_M(n-1) = \lambda B_M(n-2) + \gamma_M(n-1)\psi^2_M(n-1) \quad (14)$$

which means

$$B_M(n-1) = \min[B_{M+1}(n)]$$

Therefore, we have

$$\frac{\psi_{M+1}(n)}{\lambda B_{M+1}(n-1)}\begin{bmatrix} \mathbf{c}_{M+1}(n-1) \\ 1 \end{bmatrix} = \frac{\psi_M(n-1)}{\lambda B_M(n-2)}\begin{bmatrix} 0 \\ \mathbf{c}_M(n-2) \\ 1 \end{bmatrix} \quad (15)$$

Following the same procedure, we can prove (9). Notice that no additional computation is needed for obtaining (9) except some delays when $m > M$. This is the key point that makes the computation reduction of the BPLS algorithm possible. So the update equation for $\bar{\mathbf{k}}_N(n)$ can be written as

$$\bar{\mathbf{k}}_N(n) = \begin{bmatrix} \bar{\mathbf{k}}_M(n) \\ \mathbf{0}_{N-M} \end{bmatrix} + \sum_{i=0}^{N-M-1} \frac{\psi_M(n-i)}{\lambda B_M(n-i-1)}\begin{bmatrix} \mathbf{0}_i \\ \mathbf{c}_M(n-i-1) \\ 1 \\ \mathbf{0}_{N-M-i-1} \end{bmatrix} \quad (16)$$

In summary, when only the information of the M-th order backward predictor is available, the optimum extension of the predictor for $m > M$ satisfies the following relations

$$\psi_m(n) = \psi_M(n-m+M) \quad (17)$$

$$B_m(n) = B_M(n-m+M) \quad (18)$$

$$\mathbf{c}_m(n) = \begin{bmatrix} \mathbf{0}_{m-M} \\ \mathbf{c}_M(n-m+M) \end{bmatrix} \quad (19)$$

The extension of the conversion factor $\gamma_m(n)$, however, does not satisfy this relation, that is

$$\gamma_m(n) \neq \gamma_M(n-m+M) \quad (20)$$

This is because $\gamma_m(n)$ involves only an order-update recursion as shown in (3). There is no relation of $\gamma_m(n)$ among the time-update recursions. This fact gives the reason why the fast BPLS algorithm should be derived based on the normalized gain vector $\bar{\mathbf{k}}_m$. If the gain vector $\mathbf{k}_m$ is used, then we can write

$$\mathbf{k}_{m+1}(n) = \begin{bmatrix} \mathbf{k}_m(n) \\ 0 \end{bmatrix} + \frac{\gamma_m(n)\psi_m(n)}{B_m(n)} \begin{bmatrix} \mathbf{c}_m(n) \\ 1 \end{bmatrix} \quad (21)$$

Since (21) includes $\gamma_m(n)$, the result of (16) can not be obtained. We note that this problem was not cleared in [2].

The extension of the conversion factor can be obtained from its definition [3]

$$\gamma_m(n) = 1 - \mathbf{u}_m^T(n)\mathbf{k}_m(n) = \frac{1}{1 + \mathbf{u}_m^T(n)\bar{\mathbf{k}}_m(n)} \quad (22)$$

Multiplying both side of (16) by $\mathbf{u}_N^T(n)$ and using (1) and (22), we get

$$\frac{1}{\gamma_N(n)} = \frac{1}{\gamma_M(n)} + \sum_{i=0}^{N-M-1} \frac{\psi_M^2(n-i)}{\lambda B_M(n-i-1)} \quad (23)$$

As long as $\bar{\mathbf{k}}_N(n)$ and $\gamma_N(n)$ are available, the gain vector can be computed by $\mathbf{k}_N(n) = \gamma_N(n)\bar{\mathbf{k}}_N(n)$.

The summations on the right side of (16) and (23) can be further simplified [2]. Let $\mathbf{g}_N(n)$ and $f_N(n)$ denote the summations of (16) and (23), respectively, then we have

$$\begin{bmatrix} \mathbf{g}_N(n) \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{g}_N(n-1) \end{bmatrix} + \frac{\psi_M(n)}{\lambda B_M(n-1)} \cdot \begin{bmatrix} \mathbf{c}_M(n-1) \\ 1 \\ \mathbf{0}_{N-M} \end{bmatrix} - \frac{\psi_M(n-N+M)}{\lambda B_M(n-N+M-1)} \cdot \begin{bmatrix} \mathbf{0}_{N-M} \\ \mathbf{c}_M(n-N+M-1) \\ 1 \end{bmatrix} \quad (24)$$

and

$$f_N(n) = f_N(n-1) + \frac{\psi_M^2(n)}{\lambda B_M(n-1)} - \frac{\psi_M^2(n-N+M)}{\lambda B_M(n-N+M-1)} \quad (25)$$

The results of (24) and (25) are the same as the Version 3 of the FNTF algorithm.

The computational load of the fast BPLS algorithm is about $\frac{3}{2}M^2 + 5M + 2N$. Since $M \ll N$ is usually satisfied in some applications such as acoustic echo canceler, the computation reduction can be significant.

3 Numerical Analysis

There are many investigations concerning the numerical instability problems of the RLS algorithm and its fast versions reported in the literature. These include the unstable behavior of the conversion factor, the loss of symmetry and the loss of positive definiteness of the inverse correlation matrix of the input [4]-[7]. In [1], we have proved that these instability problems do not exist in the BPLS algorithm. In this section, we extend the conclusions to the fast BPLS algorithm.

3.1 Conversion Factor

The extension of the conversion factor from order M to N is shown by (23). Notice that the first term on the right side of (23) $1/\gamma_M(n) \geq 1$ and the second term is always greater or equal to zero. So we have $0 \leq \gamma_N(n) \leq 1$. Apparently, this result is also true in a finite-precision implementation.

However, if we use the simplified update recursion (25), then the result may not be valid under a low-bit word-length arithmetic due to the subtraction involved in (25).

3.2 Symmetric Property

Under the normal operation, the inverse correlation matrix of the input $\mathbf{P}_m(n) = \Phi_m^{-1}(n)$ should be symmetry, that is

$$\mathbf{P}_m(n) = \mathbf{P}_m^T(n) \quad (26)$$

We have shown that this property will be destroyed in the RLS algorithm but remained in the BPLS algorithm when a finite-precision arithmetic is used [1]. In order to extend this property to the fast BPLS algorithm, we use the relation $\bar{\mathbf{k}}_m(n) = \frac{1}{\lambda}\mathbf{P}_m(n-1)\mathbf{u}_m(n)$ and rewrite the summations on the right side of (16) as

$$\frac{1}{\lambda}\mathbf{P}'_N(n-1)\mathbf{u}_N(n) = \sum_{i=0}^{N-M-1} \frac{1}{\lambda B_M(n-i-1)} \begin{bmatrix} \mathbf{0}_i \\ \mathbf{c}_M(n-i-1) \\ 1 \\ \mathbf{0}_{N-M-i-1} \end{bmatrix} \cdot [\mathbf{0}_i \;\; \mathbf{c}_M(n-i-1) \;\; 1 \;\; \mathbf{0}_{N-M-i-1}] \cdot \mathbf{u}_N(n) \quad (27)$$

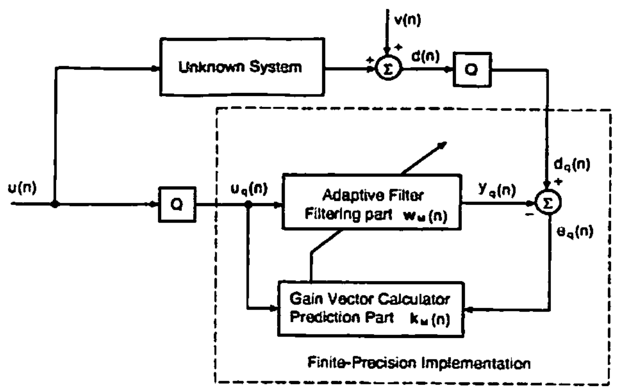

Fig. 1. Block diagram of adaptive system identification.

Apparently, $\mathbf{P}'_N(n-1) = \left(\mathbf{P}'_N(n-1)\right)^T$ is satisfied. It is not difficult to show from (27) that the symmetry is also held under a finite-precision implementation.

3.3 Positive Definiteness

The positive definiteness of $\mathbf{P}_m(n-1)$ can be defined as

$$\mathbf{u}_m^T(n)\mathbf{P}_m(n-1)\mathbf{u}_m(n) > 0 \qquad (28)$$

where $\mathbf{u}_m(n) \neq 0$ is the input vector.

Left multiplying (16) by $\mathbf{u}_N(n)$, yielding

$$\mathbf{u}_N^T(n)\mathbf{P}_N(n-1)\mathbf{u}_N(n) = \mathbf{u}_M(n)\mathbf{P}_M(n-1)\mathbf{u}_M(n) + \sum_{i=0}^{N-M-1} \frac{\psi_M^2(n-i)}{B_M(n-i-1)} \qquad (29)$$

We have proved that $\mathbf{u}_M(n)\mathbf{P}_M(n-1)\mathbf{u}_M(n) \geq 0$ [1]. Following the same procedure as shown in [1], we can prove that the summation on the right side of (29) is always greater or equal to zero. So the nonnegative definiteness of $\mathbf{P}_N(n-1)$ is guaranteed despite finite-precision implementation.

Care must be taken if we use (24), the nonnegative definiteness of $\mathbf{P}_N(n-1)$ may not be remained under a finite-precision implementation because of the subtraction involved in (24).

4 Simulation Results

To confirm the validity of our analysis and demonstrate the improved numerical performance, some simulations are carried out. An adaptive system identification problem is employed for the simulation. Its block diagram is shown in Fig.1. The blocks enclosed by the dashed line are implemented by using a floating-point arithmetic that consists of an 8-bit exponent and a variable mantissa (including a sign bit). The blocks labeled Q quantize double-precision input data into finite-precision ones that are used in the adaptive filter algorithm. A speech signal shown in Fig.2(a) is

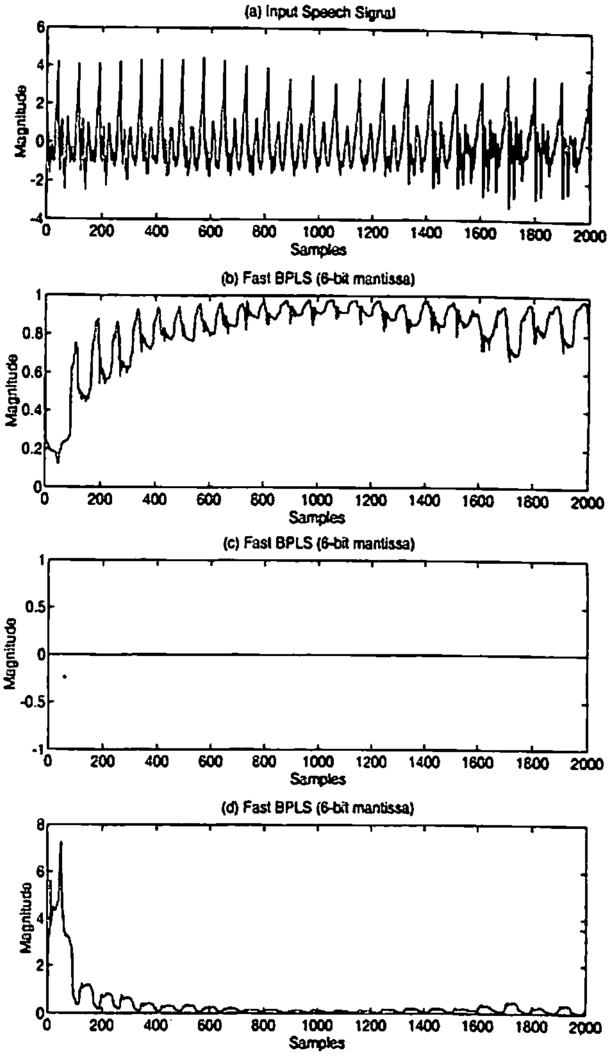

Fig. 2. Simulation conditions: $N = 50, M = 10$, 6-bit mantissa. (a) Input speech signal, (b) Conversion factor $\gamma_N^q(n)$ of fast BPLS algorithm, (c) Symmetric property of fast BPLS algorithm computed by using $\|\mathbf{P}_N^q(n-1)\mathbf{u}_N(n) - \left(\mathbf{u}_N^T(n)\mathbf{P}_N^q(n-1)\right)^T\|$. (d) Positive definiteness of fast BPLS algorithm computed by using $\mathbf{u}_N^T(n)\mathbf{P}_N^q(n-1)\mathbf{u}_N(n)$.

used as the input $u(n)$. The additive noise $v(n)$ is a white noise with zero mean. The variances of $u(n)$ and $v(n)$ are unity and 0.001, respectively. The unknown system is supposed to be a 10-th order butterworth IIR filter. The number of tap weights used in the adaptive filter is 50. The initial parameter $\delta = 1$ and the forgetting factor $\lambda = 0.98$ are used.

The simulation results of three main instability sources effects on the fast BPLS algorithm are shown in Fig.2(b)-(d). From these results, we make the following observations:

- The conversion factor in the fast BPLS algorithm is always in the range between 0 and 1 even though a low bit mantissa is used.
- The symmetric property of $\mathbf{P}_N(n-1)$ is remained in the fast BPLS algorithm.

- No loss of positive definiteness in the fast BPLS algorithm occurs under a finite precision implementation.

These observations have confirmed the validity of our analysis presented in Sec.3.

Without the effects of three main instability sources, the numerical performance of the fast BPLS algorithm is expected to be much improved. This is virtually true through computer simulations. Figure 3 shows the residual error of the fast BPLS algorithm computed by using a variety of word-length mantissa bits and compared with the FNTF algorithms. As expected, the numerical performance of the fast BPLS algorithm is very robust to round-off errors produced by finite-precision implementations. On the other hand, the FNTF combined with the fast transversal filter (FTF) algorithm is unstable even under the double-precision implementation.

5 Conclusion

A method for reducing the computational complexity of the BPLS algorithm has been proposed. The derivation is based on the assumption that the input signal can be modeled by an $AR(M)$, where M can usually be selected much smaller than the order N of the adaptive filter. The derivation is direct and simple to understand. The result is consistent with the FNTF algorithms, but the numerical performance is much improved. The improved numerical property is mainly due to the stable behavior of the conversion factor, the inherent symmetry and the guaranteed positive definiteness of the inverse correlation matrix of the input. These numerical properties have been analyzed under a finite-precision arithmetic. The computer simulation has confirmed the validity of these analyses and shown that the fast BPLS algorithm performs much more stable and robust than that of the FNTF algorithms combined with the RLS or the FRLS algorithms. Therefore, the fast BPLS algorithm can be applied to various fields, such as acoustic echo canceler, to provide a fast convergence rate and a stable numerical performance with less computation.

References

[1] Y. Wang and K. Nakayama,"Numerical performances of recursive least squares and predictor based least squares: a comparative study," *Trans. IEICE*, vol.E80-A, no.4, pp.745-752, 1997.

[2] G.V. Moustakides and S. Therodoridis,"Fast Newton transversal filters - a new class of adaptive estimation algorithms," *IEEE Trans. Signal Processing*, vol.39, no.10, pp.2184-2193, 1991.

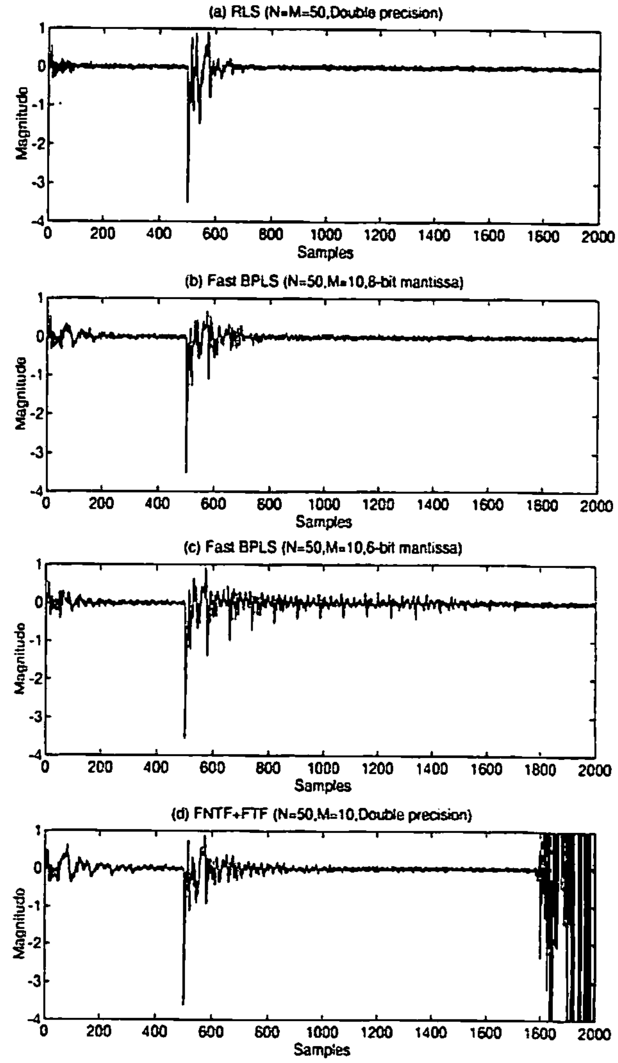

Fig. 3. Simulation conditions: $\lambda = 0.98$, $\delta = 1$, unknown system $N = 50$ and changes at 500 samples. (a) RLS algorithm, $N = M = 50$, double precision, (b) Fast BPLS algorithm, $N = 50$, $M = 10$, 8-bit mantissa, (c) Fast BPLS algorithm, $N = 50$, $M = 10$, 6-bit mantissa, (d) FNTF+FTF algorithm, $N = 50$, $M = 10$, double precision.

[3] S. Haykin, *Adaptive Filter Theory*, Second Edition, Prentice Hall, 1991.

[4] M.H. Verhaegen,"Round-off error propagation in four generally-applicable, recursive, least-squares estimation schemes," *Automatica*, vol.25, no.3 pp.437-444, 1989.

[5] G.E Bottomley and S.T. Alexander,"A theoretical basis for the divergence of conventional recursive least squares filters, " *Proc. ICASSP'89*, pp.908-911, 1989.

[6] J.M. Cioffi and T. Kailath, "Fast, recursive-least-squares transversal filters for adaptive filtering", *IEEE Trans. Acoust., Speech, Signal Processing*, vol.ASSP-32, no.2, pp.304-337, 1984.

[7] J.M. Cioffi,"Limited-precision effects in adaptive filtering, " *IEEE Trans. Circuits and Systems*, CAS-34, pp.821-833, 1987.