A numerically stable fast newton type adaptive filter based on order update fast least squares algorithm

著者	Wang Youhua, Ikeda Kazushi, Nakayama Kenji
journal or publication title	IEEE Proc. of ICASSP'98, Seattle
page range	1713-1716
year	1998-05-01
URL	http://hdl.handle.net/2297/6833

A NUMERICALLY STABLE FAST NEWTON TYPE ADAPTIVE FILTER BASED ON ORDER UPDATE FAST LEAST SQUARES ALGORITHM

Youhua WANG *Kazushi IKEDA* *Kenji NAKAYAMA*

Department of Electrical and Computer Engineering
Kanazawa University, Kanazawa 920, Japan

ABSTRACT

The numerical property of an adaptive filter algorithm is the most important problem in practical applications. Most fast adaptive filter algorithms have the numerical instability problem and the fast Newton transversal filter (FNTF) algorithms are no exception. In this paper, we propose a numerically stable fast Newton type adaptive filter algorithm. Two problems are dealt with in the paper. First, we derive the proposed algorithm from the order-update fast least squares (FLS) algorithm. This derivation is direct and simple to understand. Second, we give the stability analysis using a linear time-variant state-space method. The transition matrix of the proposed algorithm is given. The eigenvalues of the ensemble average of the transition matrix are shown to be asymptotically all less than unity. This results in a much improved numerical performance compared with the FNTF algorithms. The computer simulations implemented by using a finite-precision arithmetic have confirmed the validity of our analysis.

1. INTRODUCTION

The FNTF algorithms attract many attentions these years. The main advantage of the FNTF algorithms is the fast computation of the gain vector needed for the adaptation of the transversal filters [1]. Like any other fast version of the RLS algorithms, the FNTF algorithms also suffer from the numerical instability problem, if the predictor used for extending the gain vector is calculated by using the FLS algorithms.

The instability of the FLS algorithms is mainly produced by a hyperbolic rotation (causing the eigenvalues to go out of the unit circle) that has to be operated on the backward predictor in order to obtain the recursive equations for computing the gain vector [2]. In the FLS algorithms, however, if we assume that the recursions involve both order- and time-update, then the least squares solution can be obtained by using either forward or backward predictor. Therefore, the stable structures of both forward and backward predictors are remained. This leads to the algorithms we called the predictor based least squares (PLS) algorithms. The PLS algorithms demonstrate a very stable and robust numerical performance compared with the RLS and the FLS algorithms. Some comparative studies are presented in [3].

Unfortunately, the computational load of the PLS 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 fast PLS algorithm is proposed in this paper. The assumption for the proposed algorithm is the same as that of the FNTF, 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 Ref.[1]. Instead of using the Max-Min and Min-Max principle, the derivation shown here is direct and much easy to understand. The most important characteristic of the fast PLS algorithm is its good numerical performance. In the paper, we adopt a linear time-variant state-space method for its stability analysis. The transition matrix of the proposed algorithm is derived and its ensemble average is evaluated. The eigenvalues are shown to be asymptotically all less than unity. Computer simulations are carried out to confirm our analysis.

2. DERIVATION OF FAST PLS ALGORITHM

The derivation is based on the backward PLS (BPLS) algorithm that can be written as

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

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

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

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

 0-7803-4428-6/98 $10.00

$$\tilde{\mathbf{k}}_{m+1}(n) = \begin{bmatrix} \tilde{\mathbf{k}}_m(n) \\ 0 \end{bmatrix} + \frac{\psi_m(n)}{\lambda B_m(n-1)} \begin{bmatrix} \mathbf{c}_m(n-1) \\ 1 \end{bmatrix} \quad (5)$$

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, $\tilde{\mathbf{k}}_m(n)$ is the normalized gain vector, $\mathbf{u}_m(n)$ is the input vector and λ is the forgetting factor.

The use of the normalized gain vector $\tilde{\mathbf{k}}_m(n)$ 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 $\tilde{\mathbf{k}}_M(n)$ to $\tilde{\mathbf{k}}_N(n)$ 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

$$\frac{\psi_m(n)}{\lambda B_m(n-1)} \begin{bmatrix} \mathbf{c}_m(n-1) \\ 1 \end{bmatrix} = \frac{\psi_M(n-m+M)}{\lambda B_M(n-m+M-1)} \cdot \begin{bmatrix} \mathbf{0}_{m-M} \\ \mathbf{c}_M(n-m+M-1) \\ 1 \end{bmatrix} \quad (6)$$

To prove (6), we first compute the BPLS algorithm to get $\tilde{\mathbf{k}}_{M+1}(n)$ and the predictor of order M. Then, we write (5) for $m = M+1$ as

$$\tilde{\mathbf{k}}_{M+2}(n) = \begin{bmatrix} \tilde{\mathbf{k}}_{M+1}(n) \\ 0 \end{bmatrix} + \frac{\psi_{M+1}(n)}{\lambda B_{M+1}(n-1)} \begin{bmatrix} \mathbf{c}_{M+1}(n-1) \\ 1 \end{bmatrix} \quad (7)$$

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

$$\mathbf{c}_{M+1}(n-1) = \begin{bmatrix} 0 \\ \hat{\mathbf{c}}_M(n-1) \end{bmatrix} \quad (8)$$

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

$$\begin{aligned} \psi_{M+1}(n) &= \mathbf{c}_{M+1}^T(n-1)\mathbf{u}_{M+1}(n) + u(n-M-1) \\ &= \hat{\mathbf{c}}_M^T(n-1)\mathbf{u}_M(n-1) + u(n-M-1) \end{aligned} \quad (9)$$

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

$$\psi_M(n-1) = \mathbf{c}_M^T(n-2)\mathbf{u}_M(n-1) + u(n-M-1) \quad (10)$$

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

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

Under the constraint of using $\psi_M(n-1)$, from (2), the minimum prediction error power we can get is

$$B_M(n-1) = \lambda B_M(n-2) + \gamma_M(n-1)\psi_M^2(n-1) \quad (11)$$

which means

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

Therefore, we have

$$\frac{\psi_{M+1}(n)}{\lambda B_{M+1}(n-1)} \begin{bmatrix} \mathbf{c}_{M+1}(n-1) \\ 1 \end{bmatrix} = \frac{\psi_M(n-1)}{\lambda B_M(n-2)} \begin{bmatrix} 0 \\ \mathbf{c}_M(n-2) \\ 1 \end{bmatrix} \quad (12)$$

Following the same procedure, we can prove (6). Notice that no additional computation is needed for obtaining (6) 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 $\tilde{\mathbf{k}}_N(n)$ can be written as

$$\tilde{\mathbf{k}}_N(n) = \begin{bmatrix} \tilde{\mathbf{k}}_M(n) \\ \mathbf{0}_{N-M} \end{bmatrix} + \sum_{i=0}^{N-M-1} \frac{\psi_M(n-i)}{\lambda B_M(n-i-1)} \begin{bmatrix} \mathbf{0}_i \\ \mathbf{c}_M(n-i-1) \\ 1 \\ \mathbf{0}_{N-M-i-1} \end{bmatrix} \quad (13)$$

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

$$\gamma_m(n) \neq \gamma_M(n-m+M) \quad (14)$$

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 $\tilde{\mathbf{k}}_m(n)$. If the gain vector $\mathbf{k}_m(n)$ is used, then we can write

$$\mathbf{k}_{m+1}(n) = \begin{bmatrix} \mathbf{k}_m(n) \\ 0 \end{bmatrix} + \frac{\gamma_m(n)\psi_m(n)}{B_m(n)} \begin{bmatrix} \mathbf{c}_m(n) \\ 1 \end{bmatrix} \quad (15)$$

Since (15) includes $\gamma_m(n)$, the result of (13) can not be obtained. We note that this problem was not clarified in Ref.[1].

The extension of the conversion factor can be obtained from its definition [5]

$$\gamma_m(n) = 1 - \mathbf{u}_m^T(n)\mathbf{k}_m(n) = \frac{1}{1 + \mathbf{u}_m^T(n)\tilde{\mathbf{k}}_m(n)} \quad (16)$$

Multiplying both side of (13) by $\mathbf{u}_N^T(n)$ and using (1) and (16), we get

$$\frac{1}{\gamma_N(n)} = \frac{1}{\gamma_M(n)} + \sum_{i=0}^{N-M-1} \frac{\psi_M^2(n-i)}{\lambda B_M(n-i-1)} \quad (17)$$

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

Equations (13) and (17) can be further simplified [1], which result in the same form as the Version 3 of the FNTF algorithms.

The computational load of the fast BPLS algorithm is about $\frac{3}{2}M^2 + 5M + 2N$, which is comparable to $5M + 2N$ required by the combination of the FTF and the FNTF algorithms when M is small. This is usually satisfied in some applications like acoustic echo canceller, in which a speech signal is used as the input.

3. STABILITY ANALYSIS

The most important characteristic of the fast BPLS algorithm is its good numerical performance. In this section, we will prove this property.

The prediction part of the BPLS algorithm can be modeled by the following nonlinear state-space form [4]

$$\Theta(n) = f\left[\Theta(n-1), \mathbf{u}_m(n)\right] \quad (18)$$

where $\Theta(n)$ and $\mathbf{u}_m(n)$ denote the state-space variables and the tap-input vector, respectively. For a finite precision implementation, roundoff errors are introduced, so that $\hat{\Theta}(n) = \Theta(n) + \Delta\Theta(n)$. Assuming that the errors are small, (18) can be linearized in the presence of the roundoff errors, which leads to

$$\Delta\Theta(n) = \mathbf{A}(n)\Delta\Theta(n-1) + V(n) \quad (19)$$

where $V(n)$ represents the instantaneous contribution of the roundoff noise, and

$$\mathbf{A}(n) = \nabla_\Theta f\left[\Theta, \mathbf{u}_m(n)\right]|_{\Theta=\Theta(n-1)} \cdot \quad (20)$$

This is a linear time-variant system with a signal-dependent $\mathbf{A}(n)$ matrix. Therefore, an exact statement about the deterministic stability is difficult to make. Nevertheless, we can make certain statistical statements when the input signal $u(n)$ is stationary and ergodic. More precisely, it is shown in [4] that the state transition matrix

$$\mathbf{F}(n,0) = \mathbf{A}(n)\mathbf{A}(n-1)\cdots\mathbf{A}(1) \quad (21)$$

has an asymptotic constant eigendecomposition that can be used to decide about the numerical stability of the original system. Using the averaging technique, it is shown in [4] that numerical stability of (18) is determined by the eigenvalues of $\lim_{n\to\infty} \mathrm{E}[\mathbf{A}(n)]$.

In particular, the state-space variables that involve the time-update recursions in the BPLS algorithm can be expressed by

$$\Theta(n) = \begin{pmatrix} \mathbf{c}_m(n) \\ B_m(n) \end{pmatrix} \quad (22)$$

Substituting (1) to (4) and noticing that $\mathbf{k}_m(n) = \gamma_m(n)\tilde{\mathbf{k}}_m(n)$, we can write the first state-space variable as

$$\begin{aligned}\mathbf{c}_m(n) = (\mathbf{I}_m - \mathbf{k}_m(n)\mathbf{u}_m^T(n))\mathbf{c}_m(n-1) \\ +u(n-m)\mathbf{k}_m(n) \quad (23)\end{aligned}$$

where $\mathbf{I}_m$ is an $m \times m$ identity matrix. For the second state-space variable, substituting (1) to (2), we have

$$\begin{aligned}B_m(n) &= \lambda B_m(n-1) \\ &+\gamma_m(n)\mathbf{c}_m^T(n-1)\mathbf{u}_m(n)\mathbf{u}_m^T(n)\mathbf{c}_m(n-1) \\ &+2u(n-m)\mathbf{u}_m^T(n)\mathbf{c}_m(n-1) + u^2(n-m) \quad (24)\end{aligned}$$

It is worth noting that the gain vector $\mathbf{k}_m(n)$ and the conversion factor $\gamma_m(n)$ are not state-space variables because they involve only the order-update recursions.

From (20), the transition matrix $\mathbf{A}(n)$ is obtained by differentiating the state-space model (18) with respect to its state-space variables, which results in

$$\mathbf{A}(n) = \begin{pmatrix} \mathbf{I}_m - \mathbf{k}_m(n)\mathbf{u}_m^T(n) & 0 \\ 2\gamma_m(n)\psi_m(n)\mathbf{u}_m^T(n) & \lambda \end{pmatrix} \quad (25)$$

Since $0 < \lambda < 1$ and $\mathbf{A}(n)$ is a block-lower-triangular, it remains to show that the eigenvalues of $\mathrm{E}[\mathbf{I}_m - \mathbf{k}_m(n)\mathbf{u}_m^T(n)]$ are asymptotically all smaller than unity in magnitude.

For $1 - \lambda \ll 1$, from [5], we can derive

$$\lim_{n\to\infty} \mathbf{\Phi}_m(n) \approx (1-\lambda)^{-1}\mathbf{R} \quad (26)$$

where $\mathbf{\Phi}_m(n) = \sum_{i=1}^{n} \lambda^{n-i-1}\mathbf{u}_m(i)\mathbf{u}_m^T(i)$ and $\mathbf{R} = \mathrm{E}[\mathbf{u}_m(n)\mathbf{u}_m^T(n)]$.

Since $\mathbf{k}_m(n) = \mathbf{\Phi}_m^{-1}(n)\mathbf{u}_m(n)$, it follows that

$$\lim_{n\to\infty} \mathrm{E}[\mathbf{k}_m(n)\mathbf{u}_m^T(n)] \approx (1-\lambda)\mathbf{I}_m \quad (27)$$

Consequently,

$$\lim_{n\to\infty} \mathrm{E}[\mathbf{I}_m - \mathbf{k}_m(n)\mathbf{u}_m^T(n)] \approx \lambda\mathbf{I}_m, \quad (28)$$

which confirms that all the eigenvalues of $\mathrm{E}[\mathbf{A}(n)]$ converge approximately to λ as $n \to \infty$.

The numerical property of the fast BPLS algorithm is closely related to the BPLS algorithm. It is not difficult to see that if the backward predictor and the gain vector of order M is stable, then the extended gain vector of order N computed by (13) and (17) is also stable.

4. SIMULATION RESULTS

To confirm the validity of our analysis and demonstrate the improved numerical performance, computer simulations are carried out. An adaptive system identification problem is employed for the simulation. A floating-point arithmetic that consists of an 8-bit exponent and a variable mantissa is used for implementations.

Figure 1 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 numerically stable fast Newton type adaptive filter algorithm has been proposed. The derivation is based on the order-update FLS algorithm. The result is consistent with the FNTF algorithms, but the derivation is direct and simple to understand. The numerical property of the proposed algorithm has been analyzed by using the linear time-variant state-space method. The transition matrix of the BPLS algorithm is derived and the eigenvalues of the ensemble average of the transition matrix are are shown to be asymptotically all equal to λ. This results in a numerically stable and robust performance of the BPLS algorithm. The fast BPLS algorithm, which extend the gain vector using the parameters computed by the BPLS algorithm, behaves a much improved numerical performance compared with the FNTF algorithms. Therefore, the fast BPLS algorithm can be applied to various fields, such as acoustic echo canceller, to provide a fast convergence rate and stable performance with less computation.

References

[1] 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.

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

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

[4] D.T.M. Slock and T.Kailath,"Numerically stable fast transversal filters for recursive least squares adaptive filtering," *IEEE Trans. Signal Processing*, vol.39, no.1, pp.92-114, 1991.

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

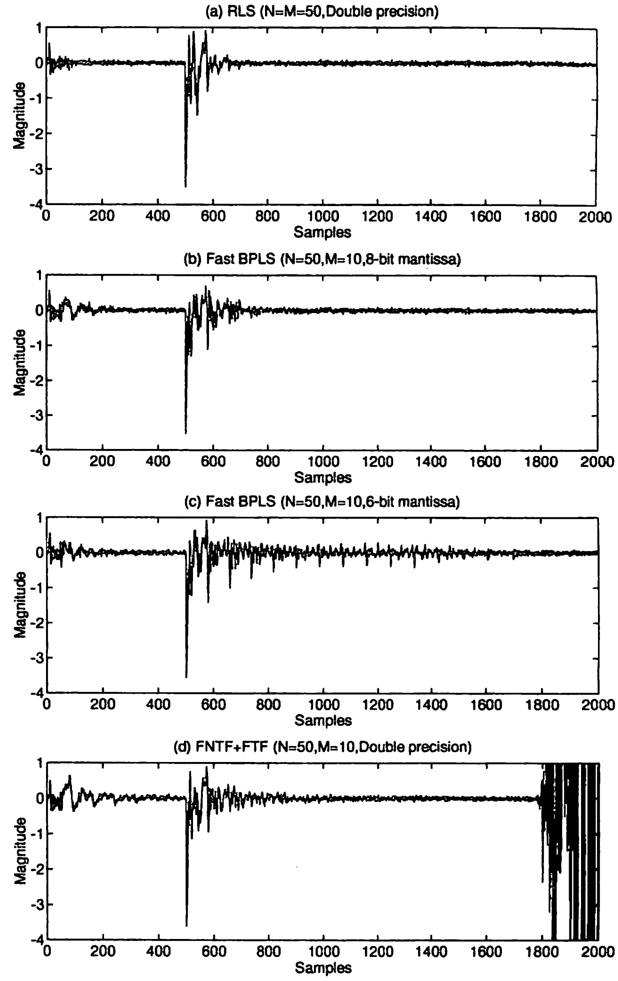

Figure 1: Simulation conditions: Speech signal input, $\lambda = 0.98$, $\delta = 1$, unknown system $N = 50$ and has a sudden change at 500 samples. (a) RLS algorithm, $N = M = 50$, double precision, (b) Fast BPLS algorithm, $N = 50$, $M = 10$, 8-bit mantissa, (c) Fast BPLS algorithm, $N = 50$, $M =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