PAPER

Numerical Performances of Recursive Least Squares and Predictor Based Least Squares: A Comparative Study

Youhua WANG† *and* Kenji NAKAYAMA†, *Members*

SUMMARY The numerical properties of the recursive least squares (RLS) algorithm and its fast versions have been extensively studied. However, very few investigations are reported concerning the numerical behavior of the predictor based least squares (PLS) algorithms that provide the same least squares solutions as the RLS algorithm. This paper presents a comparative study on the numerical performances of the RLS and the backward PLS (BPLS) algorithms. Theoretical analysis of three main instability sources reported in the literature, including the overrange of the conversion factor, the loss of symmetry and the loss of positive definiteness of the inverse correlation matrix, has been done under a finite-precision arithmetic. Simulation results have confirmed the validity of our analysis. The results show that three main instability sources encountered in the RLS algorithm do not exist in the BPLS algorithm. Consequently, the BPLS algorithm provides a much more stable and robust numerical performance compared with the RLS algorithm.

key words: *adaptive filter, RLS algorithm, fast RLS algorithm, stability, finite-precision implementation.*

1. Introduction

The recursive least squares (RLS) and the fast RLS algorithms are two well known approaches for solving the exact least squares solution in the transversal adaptive filters. Unfortunately, both algorithms suffer from the numerical instability problem under finite-precision implementations [1]–[9]. In the RLS algorithm, a well known example is the loss of symmetry and the loss of positive definiteness of the inverse correlation matrix [1]–[4]. This causes an explosive divergence. On the other hand, the instability of the fast RLS 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 [5], [6].

In the fast RLS 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) that include the forward PLS (FPLS) and the backward PLS (BPLS) algorithms.

Although the PLS algorithms can be easily derived from the fast RLS algorithms, very few investigations concerning their numerical properties are reported in the literature. In Ref. [10], we have introduced the PLS algorithms and investigated preliminarily their numerical performances. It has been shown that the PLS algorithms perform more stable than the RLS algorithm when the order of the adaptive filter is large and the forgetting factor is small. Furthermore, since the symmetric property of the input correlation matrix is exploited, the computational load of the PLS algorithms is less than 50% of that of the RLS algorithm. Nevertheless, the comparison of the numerical performance of the PLS and the RLS algorithms presented there was mainly based on the simulation results rather than theoretical analysis. Moreover, the simulations shown in Ref. [10] were carried out by using a 32-bit floating-point arithmetic, the numerical behavior of the PLS algorithms under finite-precision implementation remains unknown.

In this paper, we present a much refined comparative study on both the RLS and the PLS algorithms. Since the FPLS and the BPLS algorithms behave a very similar numerical performance, only the BPLS algorithm is dealt with in this paper. The results shown in the paper are also valid in the FPLS algorithm. First, the RLS and BPLS algorithms are given in Sect. 2 for convenience of analysis. Some parameters that have the same meanings in both algorithms are explained. In Sect. 3, a comparative study on the numerical performances of the RLS and the BPLS algorithms is addressed. Three main instability sources encountered in the RLS algorithm, including the overrange of the conversion factor, the loss of symmetry and the loss of positive definiteness of the inverse correlation matrix, are taken into account. The effects of these instability sources on both algorithms are analyzed under infinite and finite precision implementations. The validity of our study is confirmed through computer simulations in Sect. 4. A floating-point arithmetic with a variety of word-length is used for simulations. Finite-precision arithmetic error effects on both the RLS and the BPLS algorithms are investigated. Finally, the conclusion is

Manuscript received August 28, 1996.
Manuscript revised November 22, 1996.
†The authors are with the Faculty of Engineering, Kanazawa University, Kanazawa-shi, 920 Japan.

made in Sect. 5. The derivations of some results shown in Sect. 3 under a finite-precision arithmetic are given in Appendix.

2. Recursive Least Squares and Predictor Based Least Squares Algorithms

Even though the RLS and the PLS algorithms provide the same exact least square solutions, the basic principle for these two algorithms is entirely different. The RLS algorithm is derived from the so-called *matrix inversion lemma*. The PLS algorithms, however, are based on the relations between the predictors and the gain vector. Nevertheless, these two algorithms share some common features. For example, both algorithms require $O(M^2)$ computations and have the same structure that is suited for the transversal adaptive filters. This structure is featured by the tap-weight vector being updated by using the gain vector. In this section, we write these two algorithms and explain the common parameters shared by them.

The difference between the RLS and the BPLS algorithms is the method for computing the gain vector. In the RLS algorithm, the time-update recursive equations for computing the gain vector is given by

$$\gamma_M(n) = \frac{\lambda}{\lambda + \mathbf{u}_M^T(n)\mathbf{P}_M(n-1)\mathbf{u}_M(n)} \quad (1)$$

$$\mathbf{k}_M(n) = \frac{1}{\lambda}\gamma_M(n)\mathbf{P}_M(n-1)\mathbf{u}_M(n) \quad (2)$$

$$\mathbf{P}_M(n) = \frac{1}{\lambda}\Big(\mathbf{I}_M - \mathbf{k}_M(n)\mathbf{u}_M^T(n)\Big)\mathbf{P}_M(n-1) \quad (3)$$

where $\gamma_M(n)$ is the conversion factor (or the angle variable called in some references), $\mathbf{k}_M(n)$ is the gain vector, $\mathbf{P}_M(n) = \Phi_M^{-1}(n)$ is the inverse correlation matrix, $\mathbf{u}_M(n)$ is the input vector, $\mathbf{I}_M$ is the M-by-M identity matrix, λ is the forgetting factor and M is the order of the adaptive filter.

The BPLS algorithm for computing the gain vector is given by the following time- and order-update recursive equations:

$$\psi_m(n) = \mathbf{c}_m^T(n-1)\mathbf{u}_m(n) + u(n-m) \quad (4)$$

$$B_m(n) = \lambda B_m(n-1) + \gamma_m(n)\psi_m^2(n) \quad (5)$$

$$\gamma_{m+1}(n) = \frac{\lambda B_m(n-1)}{B_m(n)}\gamma_m(n) \quad (6)$$

$$\mathbf{c}_m(n) = \mathbf{c}_m(n-1) - \psi_m(n)\mathbf{k}_m(n) \quad (7)$$

$$\mathbf{k}_{m+1}(n) = \begin{bmatrix}\mathbf{k}_m(n)\\ 0\end{bmatrix} + \frac{\gamma_m(n)\psi_m(n)}{B_m(n)}\begin{bmatrix}\mathbf{c}_m(n)\\ 1\end{bmatrix} \quad (8)$$

where $\psi_m(n)$ is the backward a priori prediction error, $B_m(n)$ is the minimum power of the backward prediction error, $\gamma_m(n)$, $\mathbf{k}_m(n)$ and $\mathbf{u}_m(n)$ have the same meanings as those in the RLS algorithm but with a variable order of m.

As soon as the gain vector $\mathbf{k}_M(n)$ is available, the adaptive filtering, which is common for both algorithms, can be implemented by using the following equations:

$$\alpha(n) = d(n) - \mathbf{w}_M^T(n-1)\mathbf{u}_M(n) \quad (9)$$

$$\mathbf{w}_M(n) = \mathbf{w}_M(n-1) + \mathbf{k}_M(n)\alpha(n) \quad (10)$$

where $\alpha(n)$ is the a priori estimation error, $d(n)$ is the desired signal, $\mathbf{w}_M(n)$ is the tap-weight vector of the adaptive filter.

To initialize the RLS algorithm, we set $\mathbf{P}_M(0) = \delta\mathbf{I}_M$, δ is a small positive constant. The initial conditions for the BPLS algorithm are as follows:

At time $n = 0$, set $\mathbf{c}_m(0) = \mathbf{0}_m$, $B_m(0) = \delta$, $\mathbf{k}_m(0) = \mathbf{0}_m$ and $\gamma_m(0) = 1$, where $m = 1, 2, \cdots M-1$. At each iteration $n \geqq 1$, generate the first-order variables as follows:

$$\mathbf{k}_1(n) = \frac{u(n)}{\Phi_1(n)} \quad (11)$$

$$\gamma_1(n) = \frac{\lambda\Phi_1(n-1)}{\Phi_1(n)} \quad (12)$$

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

$$\Phi_1(n) = \lambda\Phi_1(n-1) + u^2(n) \quad (13)$$

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

It is not difficult to see that the parameters, which have the same meanings and are shared by both the RLS and the BPLS algorithms, are the conversion factor $\gamma_M(n)$ and the gain vector $\mathbf{k}_M(n)$.

3. A Comparative Study

Extensive investigations have been made concerning the numerical behaviors of various RLS algorithms [1]–[4]. These algorithms may be called differently but essentially consist of the same recursive equations as shown in Eqs. (1)-(3). Verhaegen studied several conventional RLS (CLS) algorithms and concluded that the numerical instabilities of these algorithms are mainly due to the loss of symmetry and the loss of positive definiteness of the inverse correlation matrix [2], [3]. Another instability source is the overrange of the conversion factor as reported in Ref. [4].

There are certainly other instability sources, such as stalling phenomenon which means that the filter stops adaptation, and the lack of consistent excitation of the input. These instability sources, however, may not belong to the algorithms or can be overcome by adjusting some parameters in the algorithms. For example, the stalling phenomenon can be well solved by decreasing the forgetting factor. Therefore, only three main instability sources as mentioned above are dealt with in this section. A comparative study of these instability sources in the RLS and the BPLS algorithms is presented. Both infinite and finite precision arithmetic are taken into account for the analysis.

3.1 Conversion Factor

3.1.1 RLS Algorithm

In infinite-precision arithmetic, with the inverse correlation matrix $\mathbf{P}_M(n-1)$ assumed to be positive definite, we have $\mathbf{u}_M^T(n)\mathbf{P}_M(n-1)\mathbf{u}_M(n) > 0$. According to Eq. (1), the conversion factor $\gamma_M(n)$ takes on the value between 0 and 1.

Under finite-precision implementation, however, some round-off error may be introduced in $\mathbf{P}_M(n-1)$ so that

$$\mathbf{P}_M^q(n-1) = \mathbf{P}_M(n-1) + \Delta_P(n-1). \quad (14)$$

where $(*)^q$ denotes the quantized value of $(*)$.

Substituting Eq. (14) into Eq. (1) and using the relation $\frac{1}{x+\Delta} = \frac{1}{x}\left(1+\frac{\Delta}{x}\right)^{-1} \approx \frac{1}{x}\left(1-\frac{\Delta}{x}\right) + O(\Delta^2)$, we get

$$\gamma_M^q(n) = \gamma_M(n)\big(1 - \frac{1}{\lambda}\gamma_M(n)\mathbf{u}_M^T(n)\Delta_P(n-1)\mathbf{u}_M(n)\big) + O(\Delta_P^2) \quad (15)$$

If $\Delta_P(n-1)$ loses its positive definiteness so that $\mathbf{u}_M^T(n)\Delta_P(n-1)\mathbf{u}_M(n) < 0$, then according to Eq. (15), $\gamma_M^q(n)$ may take on the value larger in magnitude than unity. On the other hand, if $\Delta_P(n-1)$ remains positive definiteness but $\mathbf{u}_M^T(n)\Delta_P(n-1)\mathbf{u}_M(n)$ is large compared to λ, it is possible now for $\gamma_M^q(n)$ to become a negative value. When these two cases happen, the RLS algorithm exhibits explosive divergence [4].

3.1.2 BPLS Algorithm

The conversion factor in the BPLS algorithm involves only an order-update recursion as shown in Eq. (6). Expanding this equation, we can write

$$\gamma_{m+1}(n) = \prod_{i=1}^{m} \frac{\lambda B_i(n-1)}{B_i(n)}\gamma_1(n) \quad (16)$$

From Eq. (5) and Eq. (12), $0 \leqq \frac{\lambda B_i(n-1)}{B_i(n)} \leqq 1$ and $0 \leqq \gamma_1(n) \leqq 1$. So the following relation can be obtained.

$$0 \leqq \gamma_{m+1}(n) \leqq \gamma_m(n) \leqq \cdots \leqq \gamma_1(n) \leqq 1 \quad (17)$$

From the derivation shown in Appendix, Eq. (17) is also valid in finite-precision implementation, Consequently, the conversion factor in the BPLS algorithm will never be negative or exceed unity.

3.2 Symmetric Property

3.2.1 RLS Algorithm

Assuming the arithmetic precision to be infinite, we have $\mathbf{u}_M^T(n)\mathbf{P}_M(n-1) = \left(\mathbf{P}_M(n-1)\mathbf{u}_M(n)\right)^T$. So the computation of Eq. (3) can be simplified. Under finite-precision implementation, however, taking advantage of this simplification will destroy the symmetric property of $\mathbf{P}_M(n)$ as analyzed in Ref. [3]. The error model, which is obtained by propagating the single error shown in Eq. (14) into Eq. (3), can be expressed as [3]

$$\begin{aligned}\Delta_P(n) = &\frac{1}{\lambda}(\mathbf{I}_M - \mathbf{k}_M(n)\mathbf{u}_M^T(n))\Delta_P(n-1) \\ &\cdot(\mathbf{I}_M - \mathbf{k}_M(n)\mathbf{u}_M^T(n))^T + \frac{1}{\lambda}(\Delta_P(n-1) \\ &-\Delta_P^T(n-1))\mathbf{u}_M(n)\mathbf{k}_M^T(n) + O(\Delta_P^2)\end{aligned} \quad (18)$$

The second term on the right side of Eq. (18) is called the non-symmetric error. It was shown that this error can be compensated by computing $\mathbf{P}_M(n-1)\mathbf{u}_M(n)$ and $\mathbf{u}_M^T(n)\mathbf{P}_M(n-1)$ separately [3]. However, even so doing, the symmetry of $\mathbf{P}_M(n)$ is not guaranteed. The reason is twofold. First, the analysis shown in Ref. [3] considered only the propagation of the single error. Second, overcoming the loss of symmetry does not mean that the loss of positive definiteness can be prevented. Experiments show that the symmetry of $\mathbf{P}_M(n)$ may not be preserved if the loss of positive definiteness occurs.

3.2.2 BPLS Algorithm

$\mathbf{P}_m(n)$ in the BPLS algorithm is inherently symmetric. This can be shown by

$$\begin{aligned}\mathbf{k}_m(n) = \mathbf{P}_m(n)\mathbf{u}_m(n) &= \left(\mathbf{k}_m^T(n)\right)^T \\ &= \left(\mathbf{u}_m^T(n)\mathbf{P}_m(n)\right)^T\end{aligned} \quad (19)$$

Apparently, Eq. (19) is also true in finite-precision implementation.

3.3 Positive Definiteness

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

$$\mathbf{u}_M^T(n)\mathbf{P}_M(n)\mathbf{u}_M(n) > 0 \quad (20)$$

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

3.3.1 RLS Algorithm

According to the definition, we multiply both sides of $\mathbf{P}_M(n)$ in Eq. (3) by $\mathbf{u}_M^T(n)$ and $\mathbf{u}_M(n)$ and rearrange terms, yielding

$$\begin{aligned}&\mathbf{u}_M^T(n)\mathbf{P}_M(n)\mathbf{u}_M(n) \\ &= \frac{1}{\lambda}\gamma_M(n)\mathbf{u}_M^T(n)\mathbf{P}_M(n-1)\mathbf{u}_M(n)\end{aligned} \quad (21)$$

Equation (21) shows that if $\mathbf{P}_M(n-1)$ is symmetric and preserve its positive definiteness, then the time-update of $\mathbf{P}_M(n)$ by using Eq. (3) remains positive definite.

Under finite-precision implementation, however, as analyzed above, $\gamma_M(n)$ may take on a negative value, leading to the negative definiteness of $\mathbf{P}_M(n)$. Generally speaking, as shown in Ref. [3] and Ref. [11], the definiteness of $\mathbf{P}_M(n)$ is closely related to the characteristics of the input signal. Assuming that an m-digit arithmetic is used, then $\mathbf{P}_M(n)$ may lose its positive definiteness if the eigenvalue spread of the inverse correlation matrix

$$\chi(\mathbf{P}_M(n)) = \frac{\lambda_{\max}}{\lambda_{\min}} > 2^m \tag{22}$$

Equation (22) gives a quite severe condition when a narrow-band signal, such as speech signal, is used as the input. The eigenvalue spread of such a signal is usually very high, which may cause the explosive divergence of the RLS algorithm.

3.3.2 BPLS Algorithm

Left multiplying Eq. (8) by $\mathbf{u}_{m+1}^T(n)$ and recognizing that $\mathbf{k}_{m+1}(n) = \mathbf{P}_{m+1}(n)\mathbf{u}_{m+1}(n)$, we get

$$\mathbf{u}_{m+1}^T(n)\mathbf{P}_{m+1}(n)\mathbf{u}_{m+1}(n) = \mathbf{u}_m^T(n)\mathbf{P}_m(n)\mathbf{u}_m(n) + \frac{\gamma_m(n)\psi_m(n)}{B_m(n)}\left(\mathbf{u}_m^T(n)\mathbf{c}_m(n) + u(n-m)\right) \tag{23}$$

From Ref. [11], the backward a posteriori prediction error $b_m(n)$ is defined as

$$b_m(n) = \mathbf{u}_m^T(n)\mathbf{c}_m(n) + u(n-m) = \gamma_m(n)\psi_m(n) \tag{24}$$

Hence, Eq. (23) can be rewritten as

$$\mathbf{u}_{m+1}^T(n)\mathbf{P}_{m+1}(n)\mathbf{u}_{m+1}(n) = \mathbf{u}_m^T(n)\mathbf{P}_m(n)\mathbf{u}_m(n) + \frac{b_m^2(n)}{B_m(n)} \tag{25}$$

Since $\frac{b_m^2(n)}{B_m(n)}$ is always positive, we can conclude that if $\mathbf{P}_m(n)$ preserves its positive definiteness, then the order-update of $\mathbf{P}_{m+1}(n)$ by using Eq. (8) remains positive definite.

To analyze the positive definiteness of $\mathbf{P}_{m+1}(n)$ under finite-precision implementation, we expand Eq. (25) as

$$\mathbf{u}_{m+1}^T(n)\mathbf{P}_{m+1}(n)\mathbf{u}_{m+1}(n) = u(n)P_1(n)u(n) + \sum_{i=1}^{m}\frac{b_i^2(n)}{B_i(n)} \tag{26}$$

In Appendix, we have proved that the terms on the right side of Eq. (26) are always greater or equal to zero. So the nonnegative definiteness of $\mathbf{P}_{m+1}(n)$ is guaranteed despite finite-precision implementation.

4. Simulation Results

Computer simulations are done to confirm the validity of our study presented in Sect. 3. An adaptive equalizer 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. The impulse response of the channel is [12]

$$h(n) = \begin{cases} \frac{1}{2}\left\{1 + \cos\left(\frac{2\pi}{W}(n-2)\right)\right\} & n = 1,2,3 \\ 0 & \text{otherwise} \end{cases} \tag{27}$$

where W controls the eigenvalue spread χ produced by the channel. We choose $W = 3.5 (\chi \approx 47)$. A speech signal shown in Fig. 2 is used as the input $u(n)$ to the channel. 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 desired signal $d(n)$ is obtained from $u(n)$ after a delay of seven samples. The adaptive equalizer has 11 taps. The RLS algorithm used for simulation is given in Sect. 2, which has the same form as CLS2 shown in Ref. [3]. The initial parameter $\delta = 0.1$ and the forgetting factor $\lambda = 0.95$ are used in both the RLS and the BPLS algorithms.

The simulation results of three main instability sources effects on the RLS and the BPLS algorithms are shown in Figs. 3, 4 and 5. From these results, we make the following observations:

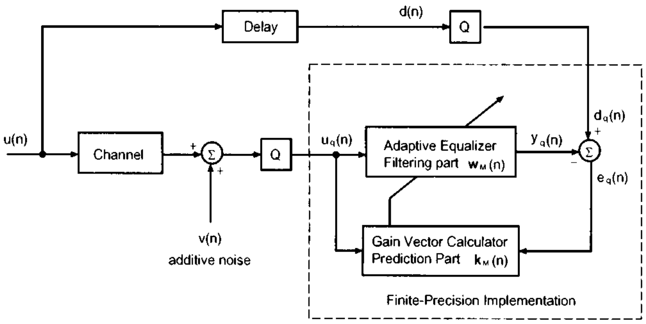


Fig. 1 Block diagram of adaptive equalizer.

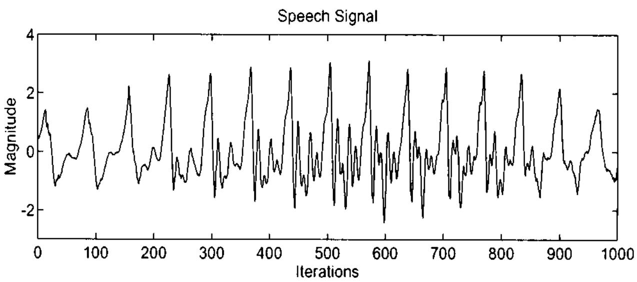


Fig. 2 Speech signal used for simulation.

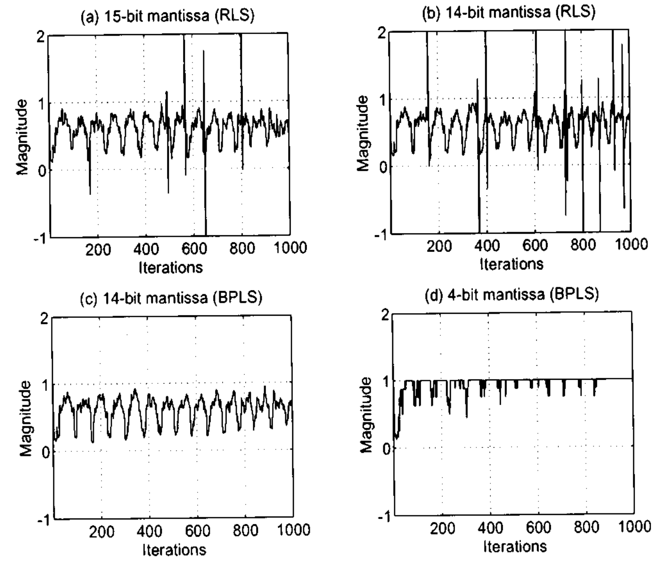


Fig. 3 Conversion factor $\gamma_M^q(n)$. (a), (b) The RLS algorithm computed by using 15-bit and 14-bit mantissa, respectively. (c), (d) The BPLS algorithm computed by using 14-bit and 4-bit mantissa, respectively.

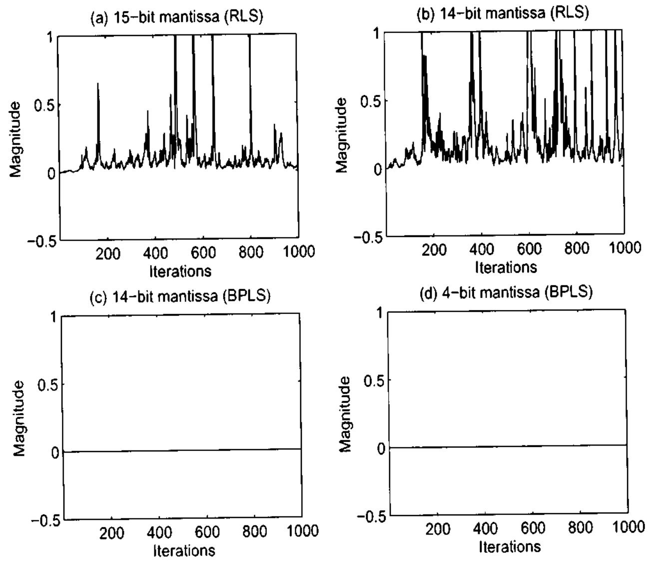


Fig. 4 Symmetric property computed by using $||\mathbf{P}_M^q(n)\mathbf{u}_M(n) - \left(\mathbf{u}_M^T(n)\mathbf{P}_M^q(n)\right)^T||$. (a), (b) The RLS algorithm computed by using 15-bit and 14-bit mantissa, respectively. (c), (d) The BPLS algorithm computed by using 14-bit and 4-bit mantissa, respectively.

- The conversion factor in the RLS algorithm may exceed the range between 0 and 1 under finite-precision implementation. No such overrange instability occurs in the BPLS algorithm.
- The symmetric property of $\mathbf{P}_M(n)$ in the RLS algorithm can not be guaranteed if the conversion factor exceeds its boundary or $\mathbf{P}_M(n)$ loses its positive definiteness. $\mathbf{P}_M(n)$ in the BPLS algorithm remains symmetry.

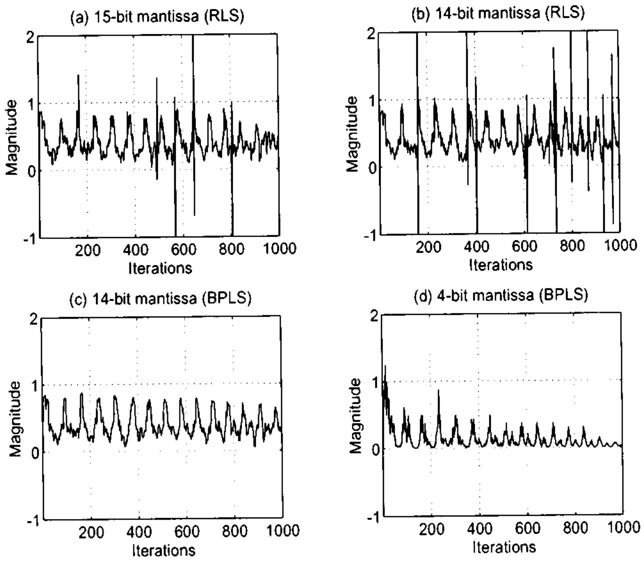


Fig. 5 Positive definiteness $\mathbf{u}_M^T(n)\mathbf{P}_M^q(n)\mathbf{u}_M(n)$. (a), (b) The RLS algorithm computed by using 15-bit and 14-bit mantissa, respectively. (c), (d) The BPLS algorithm computed by using 14-bit and 4-bit mantissa, respectively.

Table 1 MSE of BPLS and RLS algorithms.

Mantissa Bits (with a sign bit)	BPLS Algorithm	RLS Algorithm
Double precision	0.001341	0.001341
16	0.001377	0.001904
15	0.001393	unstable
8	0.001853	unstable
6	0.013333	unstable
4	0.799498	unstable

- The loss of positive definiteness in the RLS algorithm is caused by limited word-length precision, the ill conditioned input as well as the overrange of the conversion factor. No such problem has been found in the BPLS algorithm.

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

Without the effects of three main instability sources, the numerical performance of the BPLS algorithm is expected to be much more robust to round-off errors than that of the RLS algorithm. This is virtually true through computer simulations. Figures 6, 7 and Table 1 show the mean squared error (MSE) of the adaptive equalizer computed by using a variety of word-length mantissa bits. As expected, the numerical performance of the BPLS algorithm is very robust to round-off errors produced by finite-precision implementations. On the other hand, the RLS algorithm is seriously affected by finite word-length as well as the characteristic of the input signal. Simulation results show that the numerical performance of the RLS algorithm deteriorates very fast when the speech signal is used as the input. This is consistent with the analysis shown in Sect. 3. No such a deterioration has been found in

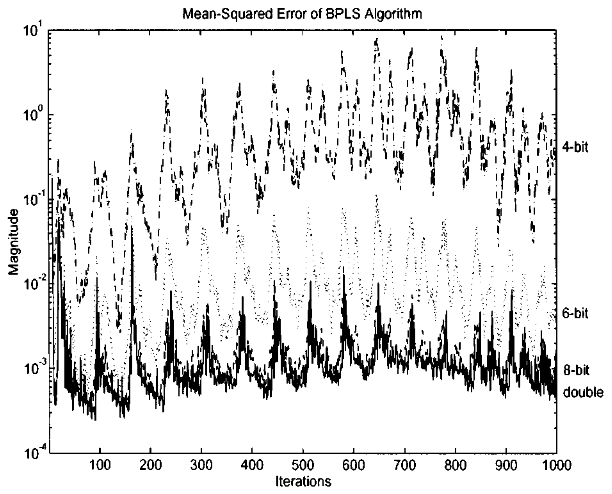


Fig. 6 Finite-precision MSE of BPLS algorithm. Double precision (solid line), 8-bit (dashed line), 6-bit (dotted line) and 4-bit (dashdot line) mantissa.

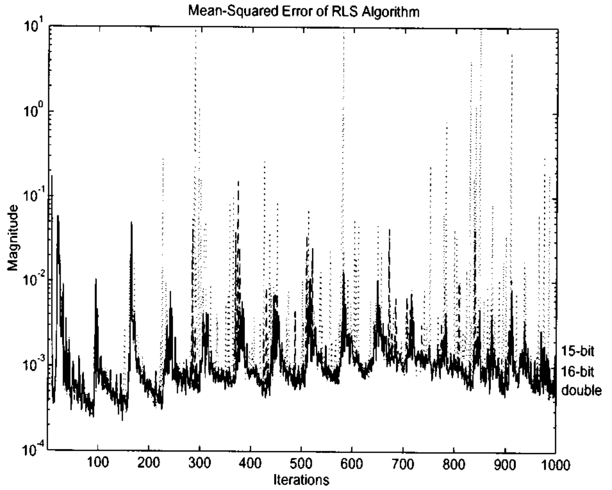


Fig. 7 Finite-precision MSE of RLS algorithm. Double precision (solid line), 16-bit (dashed line) and 15-bit (dotted line) mantissa.

the BPLS algorithm. Attention should be paid to the difference between the numerical stability and the numerical accuracy. As shown in Fig. 6, when the number of mantissa bits used to implement the BPLS algorithm is reduced, the accuracy of the algorithm is also reduced, leading to some deviations of the MSE from the ideal performance. Nevertheless, the numerical stability of the BPLS algorithm is unaffected.

5. Conclusion

A comparative study on the numerical performances of the RLS and the BPLS algorithms has been presented. Finite-precision analysis of three main instability sources, including the overrange of the conversion factor, the loss of symmetry and the loss of positive definiteness of the inverse correlation matrix, has been carried out on both the RLS and the BPLS algorithms. The validity of the analysis has been confirmed through computer simulations. It has been shown that the three main instability sources encountered in the RLS algorithm do not exist in the BPLS algorithm. This leads to a much more stable and robust numerical performance of the BPLS algorithm compared with the RLS algorithm, especially when a narrow-band signal such as the speech signal is used as the input and a low word-length finite-precision arithmetic is employed.

Acknowledgement

The authors wish to thank Mr. T. Yoshida of Matsushita Communication Kanazawa R&D Labs. Co. Ltd. for help in finite precision simulations.

References

[1] S. Ljung and L. Ljung, "Error propagation properties of recursive least-squares adaptation algorithms," Automatica, vol.21, pp.157–167, 1985.
[2] M.H. Verhaegen and P.V. Dooren, "Numerical aspects of different Kalman filter implementations," IEEE Trans. Autom. Control, vol.AC-31, no.10, 1986.
[3] 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.
[4] 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.
[5] J.M. Cioffi and T. Kailath, "Fast, recursive-least-squares transversal filters for adaptive filtering," IEEE Trans. Acoust., Speech & Signal Process., vol.ASSP-32, no.2, pp.304–337, 1984.
[6] J.M. Cioffi, "Limited-precision effects in adaptive filtering," IEEE Trans. Circuits & Syst., vol.CAS-34, pp.821–833, 1987.
[7] H. Sakai, "Survey on recent adaptive algorithms of RLS type," The Journal of the Acoustical Society of Japan, vol.48, no.7, pp.493–500, 1992.
[8] H. Schutze, "Numerical characteristics of fast recursive least squares transversal adaptation algorithms — A comparative study," Signal Processing, vol.27, pp.317–331, 1992.
[9] D.W. Lin, "On digital implementation of the fast Kalman algorithm," IEEE Trans. Acoust., Speech & Signal Process., vol.ASSP-32, no.5, pp.998–1005, 1984.
[10] Y. Wang and K. Nakayama, "A stable least square algorithm based on predictors and its application to fast newton transversal filters," IEICE Trans., vol.E78-A, no.8, pp.999–1003, 1995.
[11] R.W. Stewart and R. Chapman, "Fast stable Kalman filter algorithms utilising the square root," Proc. ICASSP'90, pp.1815–1818, 1990.
[12] S. Haykin, "Adaptive Filter Theory," Second Edition, Prentice Hall, 1991.
[13] A.V. Oppenheim and R.W. Schafer, "Digital Signal Processing," Prentice Hall, 1975.

Appendix

In finite-precision (including both floating-point and fixed-point) arithmetic, let $Q[x]$ denote the quantization of x and assume that the dynamic range for computation is large enough so that no overflow error occurs, then from Ref. [13], the following conclusions can be obtained:

(1) If $0 \leqq a \leqq 1$ and $b \geqq 0$, then $0 \leqq Q[ab] \leqq b$.

(2) If c is a real variable, then $Q[c^2] \geqq 0$.

(3) If $a \geqq b \geqq 0$ and $a \neq 0$, then $0 \leqq Q[\frac{b}{a}] \leqq 1$.

(4) If $a \geqq 0$ and $b \geqq 0$, then $Q[Q[a]+b] \geqq Q[a] \geqq 0$.

Based on the above conclusions, we prove some results shown in Sect. 3. Assume that we have the following recursive equation

$$\alpha(n) = \lambda\alpha(n-1) + \rho(n)\theta^2(n) \qquad \text{(A· 1)}$$

with $0 \leqq \lambda \leqq 1$ and $\alpha(0) = \delta > 0$. The implementation of Eq. (A· 1) under a finite-precision arithmetic is

$$\alpha^q(n) = Q\Big[Q[\lambda\alpha^q(n-1)] + Q[\rho(n)Q[\theta^2(n)]]\Big] \qquad \text{(A· 2)}$$

We want to prove that if $\rho(n) \geqq 0$, then

$$\alpha^q(n) \geqq Q[\lambda\alpha^q(n-1)] \geqq 0. \qquad \text{(A· 3)}$$

Using the mathematical induction, we first write the initial state as

$$\alpha^q(1) = Q\big[Q[\lambda\delta] + Q[\rho(1)Q[\theta^2(1)]]\big] \geqq Q[\lambda\delta] \geqq 0. \qquad \text{(A· 4)}$$

Assume

$$\alpha^q(n-1) = Q\big[Q[\lambda\alpha^q(n-2)] + Q[\rho(n-1) \cdot Q[\theta^2(n-1)]]\big] \geqq Q[\lambda\alpha^q(n-2)] \geqq 0 \qquad \text{(A· 5)}$$

then we get

$$\alpha^q(n) = Q\big[Q[\lambda\alpha^q(n-1)] + Q[\rho(n)Q[\theta^2(n)]]\big] \geqq Q[\lambda\alpha^q(n-1)] \geqq 0 \qquad \text{(A· 6)}$$

Now we prove that the conversion factor $\gamma_m(n)$ computed under finite-precision arithmetic satisfies

$$0 \leqq \gamma^q_{m+1}(n) \leqq \gamma^q_m(n) \leqq \cdots \leqq \gamma^q_1(n) \leqq 1 \qquad \text{(A· 7)}$$

First, to prove $0 \leqq \gamma^q_1(n) \leqq 1$, we write the finite-precision implementation of Eq. (13) as

$$\Phi^q_1(n) = Q\big[Q[\lambda\Phi^q_1(n-1)] + Q[u^2(n)]\big] \qquad \text{(A· 8)}$$

Apparently, Eq. (A· 8) is the special case of Eq. (A· 2) when $\rho(n) = 1$. Hence,

$$\Phi^q_1(n) \geqq Q[\lambda\Phi^q_1(n-1)] \geqq 0 \qquad \text{(A· 9)}$$

or from Eq. (12)

$$0 \leqq \gamma^q_1(n) = Q\Big[\frac{Q[\lambda\Phi^q_1(n-1)]}{\Phi^q_1(n)}\Big] \leqq 1 \qquad \text{(A· 10)}$$

Then, we use Eq. (5) and write

$$B^q_1(n) = Q\big[Q[\lambda B^q_1(n-1)] + Q[\gamma^q_1(n)Q[\psi^2_1(n)]]\big] \qquad \text{(A· 11)}$$

Notice that Eq. (A· 11) has the same form as Eq. (A· 2). So the following equation

$$0 \leqq \beta^q_1(n) = Q\Big[\frac{Q[\lambda B^q_1(n-1)]}{B^q_1(n)}\Big] \leqq 1 \qquad \text{(A· 12)}$$

is true, resulting

$$0 \leqq \gamma^q_2(n) = Q[\beta^q_1(n)\gamma^q_1(n)] \leqq \gamma^q_1(n) \leqq 1 \qquad \text{(A· 13)}$$

Following the same procedure, we can deduce that $B^q_i(n) \geqq Q[B^q_i(n-1)] \geqq 0$ and hence the final result of Eq. (A· 7).

To prove the nonnegative definiteness of $\mathbf{P}_{m+1}(n)$ in finite-precision implementation, we write Eq. (26) as

$$Q[\mathbf{u}^T_{m+1}(n)\mathbf{P}_{m+1}(n)\,\mathbf{u}_{m+1}(n)] = Q\Big[Q[u(n)P_1(n)u(n)] + \sum_{i=1}^{m} Q\Big[\frac{Q[b^2_i(n)]}{B_i(n)}\Big]\Big] \qquad \text{(A· 14)}$$

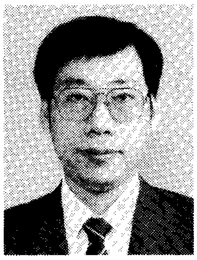
For the first term on the right side of Eq. (A· 14), from Eq. (11), we have

$$0 \leqq Q[u(n)P_1(n)u(n)] = Q\Big[\frac{Q[u^2(n)]}{\Phi^q_1(n)}\Big] \leqq 1 \qquad \text{(A· 15)}$$

For the rest terms on the right side of Eq. (A· 14), from Eqs. (24) and (5), we have

$$0 \leqq Q[b^2_i(n)] = Q[\gamma^2_i(n)\psi^2_i(n)] \leqq B^q_i(n) \qquad \text{(A· 16)}$$

From the above conclusions, we readily know that $0 \leqq Q\Big[\frac{Q[b^2_i(n)]}{B^q_i(n)}\Big] \leqq 1$, Therefore, the nonnegative definiteness of $\mathbf{P}_{m+1}(n)$ is proved.



Youhua Wang received M.S. degree from Shanghai University of Technology, China, in 1988, and Doctor of Engineering degree from Kanazawa University, Japan, in 1995. From 1988 to 1990, he held a teaching and research position at the Department of Electronic Engineering, Shenzhen University, China. From 1995 to 1996, he worked on speech coding and DSP development technology with Matsushita Communication Kanazawa R&D Laboratories. He is currently with the Department of Electrical and Computer Engineering, Kanazawa University as a lecturer. His research interests lie in the areas of digital signal processing, adaptive filtering, and their applications to communications.

Kenji Nakayama received the B.E. and Dr. degrees in electronics engineering from Tokyo Institute of Technology (TIT), Tokyo, Japan, in 1971 and 1983, respectively. From 1971 to 1972 he was engaged in the research on classical network theory in TIT. He was involved in NEC Corporation from 1972 to 1988, where his research subjects were filter design methodology and signal processing algorithms. He joined the Department of Electrical and Computer Engineering at Kanazawa University, in Aug. 1988, where he is currently a Professor. His current research interests include neural networks and adaptive filters.