THE JOURNAL OF THE ACOUSTICAL SOCIETY OF KOREA VOL.23, NO.4E 2004. 12 pp. 103~107

Partitioned Recursive Least Square Algorithm

Jun-Seok Lim*, Seokrim Choi*
*Department of Electronics Engineering, Sejong University
(Received April 8 2004; accepted November 24 2004)

Abstract

In this Paper, we propose an algorithm called partitioned recursive least square (PRLS) that involves a procedure that partitions a large data matrix into small matrices, applies RLS scheme in each of the small sub matrices and assembles the whole size estimation vector by concatenation of the sub-vectors from RLS output of sub matrices. Thus, the algorithm should be less complex than the conventional RLS and maintain an almost compatible estimation performance.

Keywords: *RLS, partitioned RLS*

I. Introduction

Recursive Least Square (RLS) is one of the most frequently employed algorithms in adaptive signal processing and control. The algorithm, however, is very complex in that it makes it difficult to apply it to a large-scale model. This is mainly due to the inherent matrix inversion and its associated multiplications and divisions. Karny and Warwick proposed a method to lessen the computational burden in a large model[1]. The method divides a large model into several low order models, estimates the partitioned low order parameters and then merges those low order models into a large model based on a general theory of pooling partial pieces of information[1]. This method needs mixing matrix to build the original model. Yu and Lee also proposed another partitioned algorithm in [2]. This algorithm has not a full recursive procedure.

In this Paper, we propose an algorithm named partitioned recursive least square (PRLS) in full recursive form, which involves a procedure that partitions a large data matrix into small matrices, applies RLS scheme in each of the small sub matrices and assembles the whole size estimation vector by concatenation of the sub-vectors from RLS output of sub matrices. Thus, by applying this procedure, the level of complexity will decrease. Here, we derive the PRLS algorithm and show the complexity reduction.

Corresponding author: Jun-Seok Lim (jslim@sejong.ac.kr)
Kwangjin-ku, Kunja-Dong 98, Sejong University, Department of Electronics Engineering, Seoul, Korea, 143-747

II. Problem Formulation

Considering the cost function as the mean square error,

$$\mathbf{J} = E\left\{\left|e(t)\right|^2\right\}, \tag{1}$$

where $e(t) = d(t) - y(t) = d(t) - \mathbf{w}^H\mathbf{u}(t)$, the optimum value of w for (1) is the solution of the following normal equation.

$$\mathbf{w} = \mathbf{R}_{\mathbf{u}}^{-1}\mathbf{P}, \tag{2}$$

where $\mathbf{R}_{\mathbf{u}} = E\left\{\mathbf{u}(t)\mathbf{u}^H(t)\right\}$ and $\mathbf{P} = E\left\{\mathbf{u}(t)d^*(t)\right\}$. When the input vector u(t) is partitioned into $\mathbf{u}(u) = \left[\mathbf{u}_1^T(t) \quad \mathbf{u}_2^T(t) \quad \mathrm{L} \quad \mathbf{u}_M^T(t)\right]$, where $\mathbf{u}_i(t)$ is an $N_i \times 1$ data vector, the autocorrelation matrix $\mathbf{R}_{\mathbf{u}}$ can be rewritten as

$$\mathbf{R}_{\mathbf{u}} = E\left\{\begin{bmatrix}\mathbf{u}_1(t)\\ \mathbf{u}_2(t)\\ \mathrm{M}\\ \mathbf{u}_M(t)\end{bmatrix}\begin{bmatrix}\mathbf{u}_1^H(t) & \mathbf{u}_2^H(t) & \mathrm{L} & \mathbf{u}_M^H(t)\end{bmatrix}\right\}$$

$$= \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} & \mathrm{L} & \mathbf{R}_{1M} \\ \mathbf{R}_{21} & \mathbf{R}_{22} & \mathrm{L} & \mathbf{R}_{2M} \\ \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} \\ \mathbf{R}_{M1} & \mathbf{R}_{M2} & \mathrm{L} & \mathbf{R}_{MM} \end{bmatrix}, \tag{3}$$

where $\mathbf{R}_{ij} = E\left\{\mathbf{u}_i(t)\mathbf{u}_j^H(t)\right\}$ for $i, j = 1,2, \ldots, M$. Let the vector $\mathbf{P}$ be correspondingly partitioned as follows:

$$\mathbf{P} = E\left\{\begin{bmatrix} \mathbf{u}_1(t) \\ \mathbf{u}_2(t) \\ \mathrm{M} \\ \mathbf{u}_M(t) \end{bmatrix} d^*(t)\right\} = \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathrm{M} \\ \mathbf{P}_M \end{bmatrix}, \tag{4}$$

where $\mathbf{P}_i = E\left\{\mathbf{u}_i(t)d^*(t)\right\}$ and the superscript * denotes the complex conjugate. Substituting (3) and (4) into (2), the optimal vector w(t) can be expressed as follows:

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathrm{M} \\ \mathbf{w}_M \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} & \mathrm{L} & \mathbf{R}_{1M} \\ \mathbf{R}_{21} & \mathbf{R}_{22} & \mathrm{L} & \mathbf{R}_{2M} \\ \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} \\ \mathbf{R}_{M1} & \mathbf{R}_{M2} & \mathrm{L} & \mathbf{R}_{MM} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathrm{M} \\ \mathbf{P}_M \end{bmatrix}, \tag{5}$$

with

$$\mathbf{w}_i = \mathbf{R}_{ii}^{-1}\left(\mathbf{P}_i - \sum_{j=1, j\neq i}^{M} \mathbf{R}_{ij}\mathbf{w}_j\right) \text{ for } i, j = 1,2, \ldots, M. \tag{6}$$

The Proposed Algorithm: For recursive formulation, the cost function can be minimized in the method of exponentially weighted least squares in place of (1),

$$\zeta(t) = \sum_{i=1}^{t} \lambda^{t-i} \left|e(i)\right|^2,$$

$$\text{where } e(i) = d(i) - y(i) = d(i) - \mathbf{w}^H(i)\mathbf{u}(i). \tag{7}$$

And then, the time update value of w(t) becomes

$$\mathbf{w}_i(t) = \mathbf{R}_{ii}^{-1}(t)\left(\mathbf{P}_i(t) - \sum_{j=1, j\neq i}^{M} \mathbf{R}_{ij}(t)\mathbf{w}_j(t)\right), \tag{8}$$

where $\mathbf{R}_{ii}(t) = \lambda\mathbf{R}_{ii}(t-1) + \mathbf{u}_i(t)\mathbf{u}_j^H(t)$ and $\mathbf{P}_i(t) = \lambda\mathbf{P}_i(t-1) + \mathbf{u}_i(t)d^*(t)$.

Assume $\Delta\mathbf{w}_j = \mathbf{w}_j(t) - \mathbf{w}_j(t-1)$, is small, (8) can be approximated as

$$\mathbf{w}_i(t) \cong \mathbf{R}_{ii}^{-1}(t)\left(\mathbf{P}_i(t) - \sum_{j=1, j\neq i}^{M} \mathbf{R}_{ij}(t)\mathbf{w}_j(t-1)\right). \tag{9}$$

Applying the matrix inversion lemma to the $\mathbf{R}_{ii}^{-1}(t)$ as

$$\mathbf{R}_{ii}^{-1}(t) = \frac{1}{\lambda}\left(\mathbf{R}_{ii}^{-1}(t-1) - \mathbf{K}_{ii}(t)\mathbf{u}_i^H(t)\mathbf{R}_{ii}^{-1}(t-1)\right), \tag{10}$$

where

$$\mathbf{K}_{ii}(t) = \frac{\mathbf{R}_{ii}^{-1}(t-1)\mathbf{u}_i(t)}{\lambda + \mathbf{u}_i^H(t)\mathbf{R}_{ii}^{-1}(t-1)\mathbf{u}_i(t)} = \mathbf{R}_{ii}^{-1}(t)\mathbf{u}_i(t),$$

the first term in (9) becomes

$$\begin{aligned} \mathbf{R}_{ii}^{-1}(t)\mathbf{P}_i(t) &= \lambda\mathbf{R}_{ii}^{-1}(t)\mathbf{P}_i(t-1) + \mathbf{R}_{ii}^{-1}(t)\mathbf{u}_i(t)d^*(t) \\ &= \mathbf{R}_{ii}^{-1}(t-1)\mathbf{P}_i(t-1) - \mathbf{K}_{ii}(t)\mathbf{u}_i^H(t)\mathbf{R}_{ii}^{-1}(t-1)\mathbf{P}_i(t-1) \\ &\quad + \mathbf{R}_{ii}^{-1}(t)\mathbf{u}_i(t)d^*(t) \end{aligned} \tag{11}$$

The second term in (9) also can be expressed as

$$\begin{aligned} &\mathbf{R}_{ii}^{-1}(t)\sum_{j=1, j\neq i}^{M} \mathbf{R}_{ij}(t)\mathbf{w}_j(t-1) = \sum_{j=1, j\neq i}^{M} \mathbf{R}_{ii}^{-1}(t)\mathbf{R}_{ij}(t)\mathbf{w}_j(t-1) \\ &= \sum_{j=1, j\neq i}^{M} \mathbf{R}_{ii}^{-1}(t)(\lambda\mathbf{R}_{ij}(t-1) + \mathbf{u}_i(t)\mathbf{u}_j^H(t))\mathbf{w}_j(t-1) \\ &= \sum_{j=1, j\neq i}^{M} \left(\mathbf{R}_{ii}^{-1}(t-1)\mathbf{R}_{ij}(t-1) - \mathbf{K}_{ii}(t)\mathbf{u}_i^H(t)\mathbf{R}_{ii}^{-1}(t-1)\mathbf{R}_{ij}(t-1)\right. \\ &\quad \left. + \mathbf{R}_{ii}^{-1}(t)\mathbf{u}_i(t)\mathbf{u}_j^H(t))\mathbf{w}_j(t-1)\right) \end{aligned} \tag{12}$$

Substituting (11) and (12) into (9), we can derive a partitioned recursive least square algorithm as (13).

$$\begin{aligned} \mathbf{w}_i(t) &= \mathbf{R}_{ii}^{-1}(t)\left(\mathbf{P}_i(t) - \sum_{j=1, j\neq i}^{M} \mathbf{R}_{ij}(t)\mathbf{w}_j(t-1)\right) \\ &= \mathbf{R}_{ii}^{-1}(t-1)\mathbf{P}_i(t-1) - \mathbf{K}_{ii}(t)\mathbf{u}_i^H(t)\mathbf{R}_{ii}^{-1}(t-1)\mathbf{P}_i(t-1) \\ &\quad + \mathbf{R}_{ii}^{-1}(t)\mathbf{u}_i(t)d^*(t) \\ &\quad - \sum_{j=1, j\neq i}^{M} \mathbf{R}_{ii}^{-1}(t-1)\mathbf{R}_{ij}(t-1)\mathbf{w}_j(t-1) \\ &\quad + \sum_{j=1, j\neq i}^{M} \mathbf{K}_{ii}(t)\mathbf{u}_i^H(t)\mathbf{R}_{ii}^{-1}(t-1)\mathbf{R}_{ij}(t-1)\mathbf{w}_j(t-1) \\ &\quad - \sum_{j=1, j\neq i}^{M} \mathbf{R}_{ii}^{-1}(t)\mathbf{u}_i(t)\mathbf{u}_j^H(t)\mathbf{w}_j(t-1) \\ &= \mathbf{R}_{ii}^{-1}(t-1)\left[\mathbf{P}_i(t-1) - \sum_{j=1, j\neq i}^{M} \mathbf{R}_{ij}(t-1)\mathbf{w}_j(t-1)\right] \\ &\quad - \mathbf{K}_{ii}(t)\mathbf{u}_i^H(t)\mathbf{R}_{ii}^{-1}(t-1)\left[\mathbf{P}_i(t-1) - \sum_{j=1, j\neq i}^{M} \mathbf{R}_{ij}(t-1)\mathbf{w}_j(t-1)\right] \end{aligned}$$

$$+\mathbf{R}_{ii}^{-1}(t)\mathbf{u}_i(t)\left[d^*(t)-\sum_{j=1,j\neq i}^{M}\mathbf{u}_j^H(t)\mathbf{w}_j(t-1)\right]$$
$$=\mathbf{w}_i(t-1)-\mathbf{K}_{ii}(t)\mathbf{u}_i^H(t)\mathbf{w}_i(t-1)$$
$$+\mathbf{K}_{ii}(t)\left[d^*(t)-\sum_{j=1,j\neq i}^{M}\mathbf{u}_j^H(t)\mathbf{w}_j(t-1)\right] \quad (13)$$

where

$$\mathbf{K}_{ii}(t)=\frac{\mathbf{R}_{ii}^{-1}(t-1)\mathbf{u}_i(t)}{\lambda+\mathbf{u}_i^H(t)\mathbf{R}_{ii}^{-1}(t-1)\mathbf{u}_i(t)}$$

and $\mathbf{R}_{ii}^{-1}(t)=\frac{1}{\lambda}\left(\mathbf{R}_{ii}^{-1}(t-1)-\mathbf{K}_{ii}(t)\mathbf{u}_i^H(t)\mathbf{R}_{ii}^{-1}(t-1)\right)$.

$$\mathbf{w}_i(t-1)-\mathbf{K}_{ii}(t)\mathbf{u}_i^H(t)\mathbf{w}_i(t-1)$$
$$+\mathbf{K}_{ii}(t)\left[d^*(t)-\sum_{j=1,j\neq i}^{M}\mathbf{u}_j^H(t)\mathbf{w}_j(t-1)\right]$$

III. Computational Complexity

Assuming the weight vector for full system is **w(t)**, the weight vector in the ith subsystem is represented by $\mathbf{w}_i$, whose size is N_i of N/M. We can derive total number of multiplication from eqn. (13). $\mathbf{K}_{ii}(t)$ needs $N_i^2+N_i$ multiplications. $\mathbf{R}_{ii}^{-1}(t)$ does N_i^2 because $\mathbf{R}_{ii}^{-1}(t)$ has common part with $\mathbf{K}_{ii}(t)$. $\mathbf{K}_{ii}(t)\mathbf{u}_i^H(t)\mathbf{w}_i(t-1)$ does $2N_i$.

$\mathbf{K}_{ii}(t)\left[d^*(t)-\sum_{j=1,j\neq i}^{M}\mathbf{u}_j^H(t)\mathbf{w}_j(t-1)\right]$ does $(M-1)N_i+N_i$.

Therefore, the total number of multiplications required is

$$M\left(2N_i^2+(M+3)N_i\right)$$
$$=M\left(2\left(\frac{N}{M}\right)^2+(M+3)\frac{N}{M}\right)=\frac{2}{M}N^2+(M+3)N \quad (14)$$

where N is the length of weight vector **w**(t) or data vector **u**(t). In contrast, conventional RLS algorithm requires about $2N^2+4N$ [3]. To compare the required computational complexities between the proposed and conventional method, the following inequality equation is solved.

$$\frac{2}{M}N^2+(M+3)N<2N^2+4N. \quad (15)$$

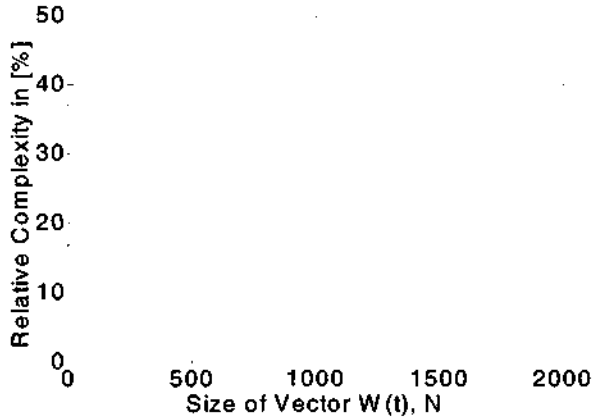

Fig. 1. Relative complexity of PRLS to conventional RLS in optimal partition

From (15), we achieve the valid range of M as follows:

$$1<M<2N. \quad (16)$$

In actual fact, the partition number M should be less than the data vector length of N such that the proposed algorithm always needs lesser multiplications than the conventional RLS. We can also derive the optimal partition number from differentiation by M of (14) in case of equal division as follows.

Optimal partition number: $M=\sqrt{2N}$. (17)

Fig. 1. shows the relative complexity of the ratio between the proposed algorithm and the conventional RLS in percent in optimal partition. Fig. 1. illustrates that relative complexity of PRLS decreases as the order of whole system increases.

IV. Simulation Results

In this section, a system identification example is presented for performance comparison. We considered an FIR model such as

$$y(t)=\mathbf{h}^T\mathbf{x}=\begin{bmatrix}-0.3 & -0.9 & 0.8 & -0.7 & 0.6 & 0.1\end{bmatrix}\begin{bmatrix}x(t)\\ \text{M}\\ x(t-5)\end{bmatrix} \quad (18)$$

Table 1. Performance comparison between conventional RLS and PRLS in a small system case

SNR [dB]	RLS	PRLS (2 subsystems)	PRLS (3 subsystems)
	RMSE[1)]	RMSE	RMSE
5	3.0 %	3.0 %	3.1 %
10	1.7 %	1.7 %	1.6 %
15	0.95 %	0.96 %	0.95 %
20	0.51 %	0.53 %	0.50 %
25	0.29 %	0.29 %	0.27 %
R.C.[2)]	100%	75%	69%

1) RMSE: normalized root mean square error

$$\text{RMSE} = \sqrt{\frac{1}{M}\sum_{m=1}^{M}\frac{\|\hat{\mathbf{h}}_m - \mathbf{h}\|^2}{\|\mathbf{h}\|^2}}$$

$\hat{\mathbf{h}}_m$: the estimated vector in the m*th* trial

2) R.C.: relative complexity (relative number of multiplication)

Table 2. Performance comparison between conventional RLS and PRLS in a small system case

SNR [dB]	RLS	PRLS (5 subsystems)	PRLS (10 subsystems)	PRLS (20 subsystems)	PRLS (50 subsystems)
	RMSE[1)]	RMSE	RMSE	RMSE	RMSE
10	7.28%	7.87%	7.96%	8.02%	7.96%
15	4.08%	4.36%	4.42%	4.51%	4.59%
20	2.31%	2.49%	2.53%	2.55%	2.54%
25	1.30%	1.40%	1.40%	1.43%	1.45%
R.C.[2)]	100%	23%	16%	16%	28%

1) RMSE: normalized root mean square error

$$\text{RMSE} = \sqrt{\frac{1}{M}\sum_{m=1}^{M}\frac{\|\hat{\mathbf{h}}_m - \mathbf{h}\|^2}{\|\mathbf{h}\|^2}}$$

$\hat{\mathbf{h}}_m$: the estimated vector in the m*th* trial

2) R.C.: relative complexity (relative number of multiplication)

where x(t) is white noise. In this simulation, we applied the proposed algorithm to the model, **h**, partitioned by 2 and 3 subsystems, and compared the results from conventional RLS with the whole length system of **h**. Table 1 gives the summary of the results from 100 independent trials.

In Table 1, the proposed algorithm shows almost the level of estimation accuracy with little degradation based on RMSE. The second model has 100 taps, with nonzero taps of values [0.1, 1, -0.5, 0.1] located in positions [1, 30, 35, 85]. In this model, we divide it into 5, 10, 20 and 50 subsystems respectively. Table 2 summarizes the results from 100 independent trials. It shows the same performance as a former case. Fig.2. compares the MSE (mean square error). It contains result of RLS and results of the proposed algorithm with partitioning in 5, 10, 20 and 50. This figure shows that the proposed algorithm converges almost at the same speed although the proposed one converges slower at the early phase. In this simulation, we add white Gaussian noise to clean output, y, for the various SNRs.

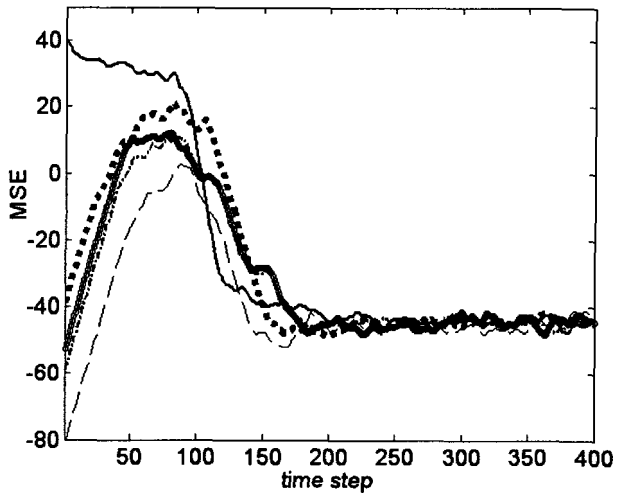

Fig. 2. MSE comparison between the proposed algorithm and the conventional RLS
(solid: RLS, dash: proposed one with 5 partitions, dot: proposed one with 10 partitions, dash and dot: proposed one with 20 partitions, circle: proposed one with 50 partitions)

V. Conclusion

In this Paper, we proposed an efficient RLS algorithm for a large order model. The algorithm decomposes the parameter vector of the original system into several subvectors and applies each one to independent RLS. The whole size parameter vector can be constructed merely by concatenation of the output subvectors. The positive aspect of this method is that it is less complex compared to the conventional RLS. The comparison also shows that the same estimation performance is maintained.

Acknowledgements

This result was supported by UARC and ADD.

References

1. M.Karny and K. Warwick, "System identification using partitioned least squares," IEE Proc.-Control Theory Appli., 142(3), May 1995, pp.223~228
2. S. J. Yu and J. H. Lee, "Adaptive array beamforming based on an efficient technique," IEEE Trans. On Antennas and Propagation, 44(8), Aug. 1996, pp. 1094~1101
3. S. Haykin, *Adaptive Filter Theory ed. 4*, (Prentice-Hall, New York, 1998)

[Profile]

• Jun-Seok Lim

Ph.D., Seoul National Univ. 1996

M.S., Seoul National Univ. 1988

B.S., Seoul National Univ. 1986

Research Interest: Array Signal Processing, Acoustic Signal Processing.

He has worked in Sejong University as associate professor since 1998.

• Seokrim Choi

Ph.D.: Computer Vision, Syracuse University, New York, USA. 12/1992

M.S.: Electronic Engineering, Seoul National University, Seoul, Korea, 2/1983

B.S.: Electronic Engineering, Seoul National University, Seoul, Korea, 2/1981

Research Interest: Image Communication, Image Compression, MPEG, Multimedia.

He has worked in Sejong University as associate professor since 1997.