THE JOURNAL OF THE ACOUSTICAL SOCIETY OF KOREA Vol.22, No.3E 2003. 9. pp. 93~100

Adaptive System Identification Using an Efficient Recursive Total Least Squares Algorithm

Nakjin Choi*, Jun-Seok Lim**, Joon-Il Song*, Koeng-Mo Sung*
*School of Electrical Engineering, Seoul National University
**Department of Electronics Engineering, Sejong University
(Received April 15 2002; revised August 1 2003; accepted August 18 2003)

Abstract

We present a recursive total least squares (RTLS) algorithm for adaptive system identification. So far, recursive least squares (RLS) has been successfully applied in solving adaptive system identification problem. But, when input data contain additive noise, the results from RLS could be biased. Such biased results can be avoided by using the recursive total least squares (RTLS) algorithm. The RTLS algorithm described in this paper gives better performance than RLS algorithm over a wide range of SNRs and involves approximately the same computational complexity of $O(N^2)$.

Keywords: *TLS, Recursive total least squares, System identification*

I. Introduction

Since the late 1950's, a number of researchers have developed a wide variety of adaptive filtering algorithms. Among these algorithms, the recursive least squares (RLS) algorithm has been used in numerous signal processing applications including adaptive channel equalization, speech coding, spectrum analysis, adaptive noise cancellation and adaptive beamforming[1]. The RLS algorithm is based on the least squares (LS) method which solves the linear equation $\mathbf{Ax} = \mathbf{b}$ as $\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$. In the LS problem, the underlying assumption is that we know the data matrix A exactly and all the errors are confined to the observation vector **b**. Unfortunately, this assumption is frequently not true. Because sampling errors, human errors, modeling errors, and instrument errors may preclude the possibility of knowing the data matrix **A** exactly[2].

Corresponding author: Nakjin Choi (nakjin@acoustics.snu.ac.kr)
School of Electrical Engineering, Seoul National University, San 56-1, Shillim-dong, Kwanak-gu, 151-742, Korea

To overcome this problem, the total least squares (TLS) method has been devised. This method compensates for the errors of data matrix **A** and the errors of observation vector **b**, simultaneously. TLS was independently derived in several areas of science. There has been several references made to TLS as early as 1901[3]. However, in the field of numerical analysis, the TLS problem was first introduced by Golub and Van Loan in 1980[2], and was extensively studied and refined by Van Huffel, Vandevalle, Lemmerling, Hansen and O'Leary, as well as many other researchers[4]. Most N-dimensional TLS solutions have been obtained by computing a singular value decomposition (SVD), generally requiring $O(N^3)$ multiplications. To reduce this computational complexity, in particular, Davila, Cichocki, Gao, Chen along with many others have developed and analyzed adaptive algorithms that employ the TLS formulation and their extensions[5-8]. These algorithms can be categorized into two groups: the first stochastic gradient type algorithms and the second least squares type algorithm.

In this paper, we present a new recursive TLS algorithm by applying a different approach. This algorithm recursively calculates and tracks the eigenvector corresponding to the minimum eigenvalue, the TLS solution, from the inverse correlation matrix of the augmented sample matrix. Then, we demonstrate that this algorithm outperforms the RLS algorithm in adaptive system identification through computer simulation results. Moreover, we show that the recursive TLS algorithm involves approximately the same order of computational complexity $O(N^2)$ as RLS algorithm.

In Section II of this paper, we derive a new recursive TLS algorithm and in Section III, we explain how to apply this algorithm to adaptive system identification. In Section IV, the results and analysis of our experiment are given followed by our conclusions in Section V.

II. Derivation of a New RTLS Algorithm

The TLS problem is a minimization problem described as follows.

$$\underset{\mathbf{E},\mathbf{r}}{\text{minimize}} \|[\mathbf{E} \mid \mathbf{r}]\|_F \quad \text{subject to} \quad (\mathbf{A}+\mathbf{E})\mathbf{x} = \mathbf{b}+\mathbf{r} \qquad (1)$$

where $\mathbf{A}$ is an M×N input matrix, $\mathbf{b}$ is an M×1 output vector. $\mathbf{E}$, $\mathbf{r}$ are the M×N input error matrix and M×1 output error vector respectively. Here, $\|\cdot\|_F$ denotes the Frobenius norm, that is, $\|B\|_F^2 = \sum_i\sum_j |b_{ij}|^2$. Once a minimization $\mathbf{E}$, $\mathbf{r}$ are found, then any $\mathbf{x}$ satisfying

$$(\mathbf{A}+\mathbf{E})\mathbf{x} = \mathbf{b}+\mathbf{r} \qquad (2)$$

is assumed to be able to solve the TLS problem in eq. (1). We denote this solution as $\mathbf{x}_{TLS}$.

Let's define the M×(N+1) augmented data matrix $\overline{\mathbf{A}}$ as follows.

$$\overline{\mathbf{A}} = [\mathbf{A} \mid \mathbf{b}] \qquad (3)$$

where $\mathbf{A} = \begin{bmatrix} \mathbf{a}_1^H \\ \vdots \\ \mathbf{a}_M^H \end{bmatrix}$, $\mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_M \end{bmatrix}$ and $\overline{\mathbf{A}} = \begin{bmatrix} \overline{\mathbf{a}}_1^H \\ \vdots \\ \overline{\mathbf{a}}_M^H \end{bmatrix}$

If we perform SVD (Singular Value Decomposition) for $\overline{\mathbf{A}}$, then we can express $\overline{\mathbf{A}}$ as follows.

$$\overline{\mathbf{A}} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^H$$

$$= \begin{bmatrix} | & & | \\ \mathbf{u}_1 & \cdots & \mathbf{u}_M \\ | & & | \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{N+1} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} | & & | \\ \mathbf{v}_1 & \cdots & \mathbf{v}_{N+1} \\ | & & | \end{bmatrix}^H \qquad (4)$$

Golub and Van Loan proved that the TLS solution $\mathbf{x}_{TLS}$ involves the most right singular vector $\mathbf{v}_{N+1}$ of the $\overline{\mathbf{A}}$ as follows[2].

$$\mathbf{x}_{TLS} = -\frac{1}{v_{N+1,N+1}} \begin{bmatrix} v_{N+1,1} \\ \vdots \\ v_{N+1,N} \end{bmatrix} \qquad (5)$$

where $\mathbf{v}_{N+1} = \begin{bmatrix} v_{N+1,1} \\ \vdots \\ v_{N+1,N+1} \end{bmatrix}$

Now we develop the adaptive algorithm which calculates the TLS solution recursively. First, n×(N+1) augmented data matrix $\overline{\mathbf{A}}(n)$ can be defined by

$$\overline{\mathbf{A}}(n) = \begin{bmatrix} \overline{\mathbf{a}}_1^H \\ \vdots \\ \overline{\mathbf{a}}_n^H \end{bmatrix} = \mathbf{U}(n)\boldsymbol{\Sigma}(n)\mathbf{V}^H(n) \qquad (6)$$

$$= \begin{bmatrix} | & & | \\ \mathbf{u}_1(n) & \cdots & \mathbf{u}_M(n) \\ | & & | \end{bmatrix} \begin{bmatrix} \sigma_1(n) & 0 & \cdots & 0 \\ 0 & \sigma_2(n) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{N+1}(n) \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} | & & | \\ \mathbf{v}_1(n) & \cdots & \mathbf{v}_{N+1}(n) \\ | & & | \end{bmatrix}^H$$

Then, we use Eqs. (5) and (6) to obtain a recursive TLS solution $\mathbf{v}_{N+1}(n)$. We know that $\mathbf{v}_{N+1}(n)$ can be obtained from the SVD of the inverse correlation matrix $\mathbf{P}(n)$ which is defined as follows (See Appendix A).

$$\mathbf{P}(n) = \mathbf{R}^{-1}(n) \text{ where } \mathbf{R}(n) = \overline{\mathbf{A}}^H(n)\overline{\mathbf{A}}(n) \quad n=1,2,\ldots, M \qquad (7)$$

Now, we derive the adaptive algorithm which recursively calculates $\mathbf{v}_{N+1}(n)$. From now on, we call the eigenvector $\mathbf{v}_{N+1}(n)$ corresponding to the minimum eigenvalue as minimum eigenvector. The minimum eigenvector $\mathbf{v}_{N+1}(n\text{-}1)$ at time index n-1 can be represented as a linear combination of the orthogonal eigenvectors $\mathbf{v}_1(n), \ldots, \mathbf{v}_{N+1}(n)$ at time index n.

$$\mathbf{v}_{N+1}(n-1) = c_1(n)\mathbf{v}_1(n) + \cdots + c_{N+1}(n)\mathbf{v}_{N+1}(n) \tag{8}$$

Because $\mathbf{P}(n)$ and $\mathbf{P}(n\text{-}1)$ are highly correlated, the coefficient $c_{N+1}(n)$ to which the minimum eigenvector $\mathbf{v}_{N+1}(n)$ is multiplied, produces a larger value than any other coefficients. That is,

$$c_{N+1}(n) \geq c_N(n) \geq \cdots \geq c_1(n) \tag{9}$$

Now, consider the new vector $\mathbf{v}(n)$ which is defined as follows.

$$\mathbf{v}(n) = \mathbf{P}(n)\mathbf{v}_{N+1}(n-1) \tag{10}$$

By using the Appendix A and Eq. (8), we can rewrite the vector $\mathbf{v}(n)$ as Eq. (11)

$$\begin{aligned}\mathbf{v}(n) &= \mathbf{P}(n)\mathbf{v}_{N+1}(n-1) \\ &= \left(\sum_{i=1}^{N+1} \frac{1}{\sigma_i^2(n)} \mathbf{v}_i(n)\mathbf{v}_i^H(n)\right)\left(\sum_{j=1}^{N+1} c_j(n)\mathbf{v}_j(n)\right) \\ &= \sum_{i=1}^{N+1} \frac{c_i(n)}{\sigma_i^2(n)} \mathbf{v}_i(n)\end{aligned} \tag{11}$$

where $\sigma_1^2(n) \geq \sigma_2^2(n) \geq \cdots \geq \sigma_{N+1}^2(n)$

Combining the inequalities in Eqs. (9), (11), the order of the magnitudes of the coefficients $\mathbf{v}_i(n)$ becomes

$$\frac{c_{N+1}(n)}{\sigma_{N+1}^2(n)} \geq \frac{c_N(n)}{\sigma_N^2(n)} \geq \cdots \geq \frac{c_1(n)}{\sigma_1^2(n)} \tag{12}$$

Because of the above reasons, the (N+1)-th term in Eq. (11) produces a larger value than any other terms. So $\mathbf{v}(n)$ converges to the scaled minimum eigenvector as follows.

$$\mathbf{v}(n) \approx \frac{c_{N+1}(n)}{\sigma_{N+1}^2(n)} \mathbf{v}_{N+1}(n) \tag{13}$$

Therefore, the minimum eigenvector $\mathbf{v}_{N+1}(n)$ can be approximately calculated from the $\mathbf{v}(n)$ as in Eq. (14).

$$\hat{\mathbf{v}}_{N+1}(n) = \frac{\mathbf{v}(n)}{\|\mathbf{v}(n)\|} \tag{14}$$

Finally, the estimate of the TLS solution at the time index n is as follows.

$$\hat{\mathbf{x}}_{TLS}(n) = -\frac{1}{\hat{v}_{N+1,N+1}(n)} \begin{bmatrix} \hat{v}_{N+1,1}(n) \\ \vdots \\ \hat{v}_{N+1,N}(n) \end{bmatrix} \tag{15}$$

where $\hat{\mathbf{v}}_{N+1}(n) = \begin{bmatrix} \hat{v}_{N+1,1}(n) \\ \vdots \\ \hat{v}_{N+1,N+1}(n) \end{bmatrix}$

The above procedure is summarized in Table 1.

As summarized above, $\mathbf{P}(n)$ is updated at each time n by using the matrix inversion lemma[1], which is used to calculate the approximate TLS solution $\hat{\mathbf{x}}_{TLS}(n)$. This new RTLS algorithm requires almost the same order of multiplications $O(N^2)$. Moreover, it maintains the structure of the conventional RLS algorithm so that the complexity can be reduced by adopting many kinds of fast algorithms for RLS. We also summarized the RLS algorithm and its computational complexity in Appendix B.

III. TLS for Adaptive System Identification

The purpose of the adaptive system identification is to estimate the impulse response of an unknown system based on noisy measurements of the system's input and output.

Consider the adaptive system identification model as depicted in Figure 1.

Here, the unknown system impulse response is assumed to be the N×1 vector

$$\mathbf{x}^* = \left[x_0^* \ \cdots \ x_{N-1}^*\right]^T \tag{16}$$

In the noise-free unknown system, the relationship between the input vector $\mathbf{a}^*(n)$ and the output value $b^*(n)$ is

Table 1. Summary of the new RTLS algorithm.

Initialize this algorithm by setting $$\mathbf{P}(0)=\delta^{-1}\mathbf{I},\quad \hat{\mathbf{v}}_{N+1}(0)=\frac{\mathbf{1}}{\lVert\mathbf{1}\rVert}$$ where δ is a small positive constant, $\mathbf{I}$ is an (N+1)×(N+1) identity matrix and $\mathbf{1}$ is an (N+1)×1 1's vector.	
Compute for each instant of time, n=1,2, ...,M	MAD's
1. $\bar{\mathbf{a}}(n)=\begin{bmatrix}\mathbf{a}(n)\\ b(n)\end{bmatrix}$	
2. $\mathbf{k}(n)=\dfrac{\lambda^{-1}\mathbf{P}(n-1)\bar{\mathbf{a}}(n)}{1+\lambda^{-1}\bar{\mathbf{a}}^{H}(n)\mathbf{P}(n-1)\bar{\mathbf{a}}(n)}$	2. N^2+4N+4
3. $\mathbf{P}(n)=\lambda^{-1}\mathbf{P}(n-1)-\lambda^{-1}\mathbf{k}(n)\bar{\mathbf{a}}^{H}(n)\mathbf{P}(n-1)$	3. $3N^2+6N+3$
4. $\mathbf{v}(n)=\mathbf{P}(n)\hat{\mathbf{v}}_{N+1}(n-1)$	4. N^2+2N+1
5. $\hat{\mathbf{v}}_{N+1}(n)=\dfrac{\mathbf{v}(n)}{\lVert\mathbf{v}(n)\rVert}$	5. N^2+3N+3
6. $\hat{\mathbf{x}}_{TLS}(n)=-\dfrac{1}{\hat{v}_{N+1,N+1}(n)}\begin{bmatrix}\hat{v}_{N+1,1}(n)\\ \vdots\\ \hat{v}_{N+1,N}(n)\end{bmatrix}$ where $\hat{\mathbf{v}}_{N+1}(n)=\begin{bmatrix}\hat{v}_{N+1,1}(n)\\ \vdots\\ \hat{v}_{N+1,N+1}(n)\end{bmatrix}$	6. $N+1$
Total number of MAD's	$6N^2+16N+12$

MAD's stands for number of multiplies, divides, and square roots.

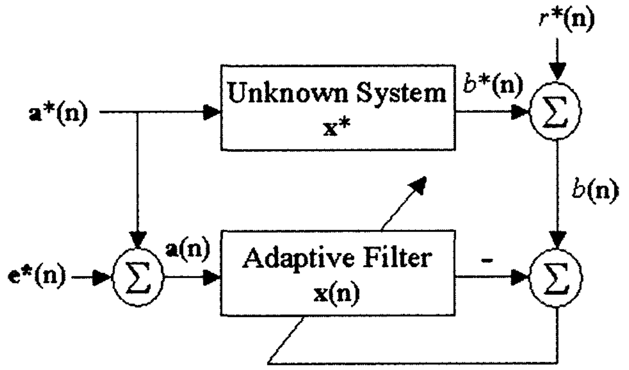


Figure 1. Adaptive system identification.

$$(\mathbf{a}^*(n))^T\mathbf{x}^*=b^*(n) \tag{17}$$

where $\mathbf{a}^*(n)=[a_0^*(n)\ \cdots\ a_{N-1}^*(n)]^T$

But in the real environment, the input vector $\mathbf{a}(n)$ and the output value $b(n)$ contain additive noise. The relationship between $\mathbf{a}(n)$ and $b(n)$ is

$$(\mathbf{a}(n))^T\mathbf{x}(n)\approx b(n) \tag{18}$$

where $\mathbf{a}(n)=\mathbf{a}^*(n)+\mathbf{e}^*(n)$ and $b(n)=b^*(n)+r^*(n)$

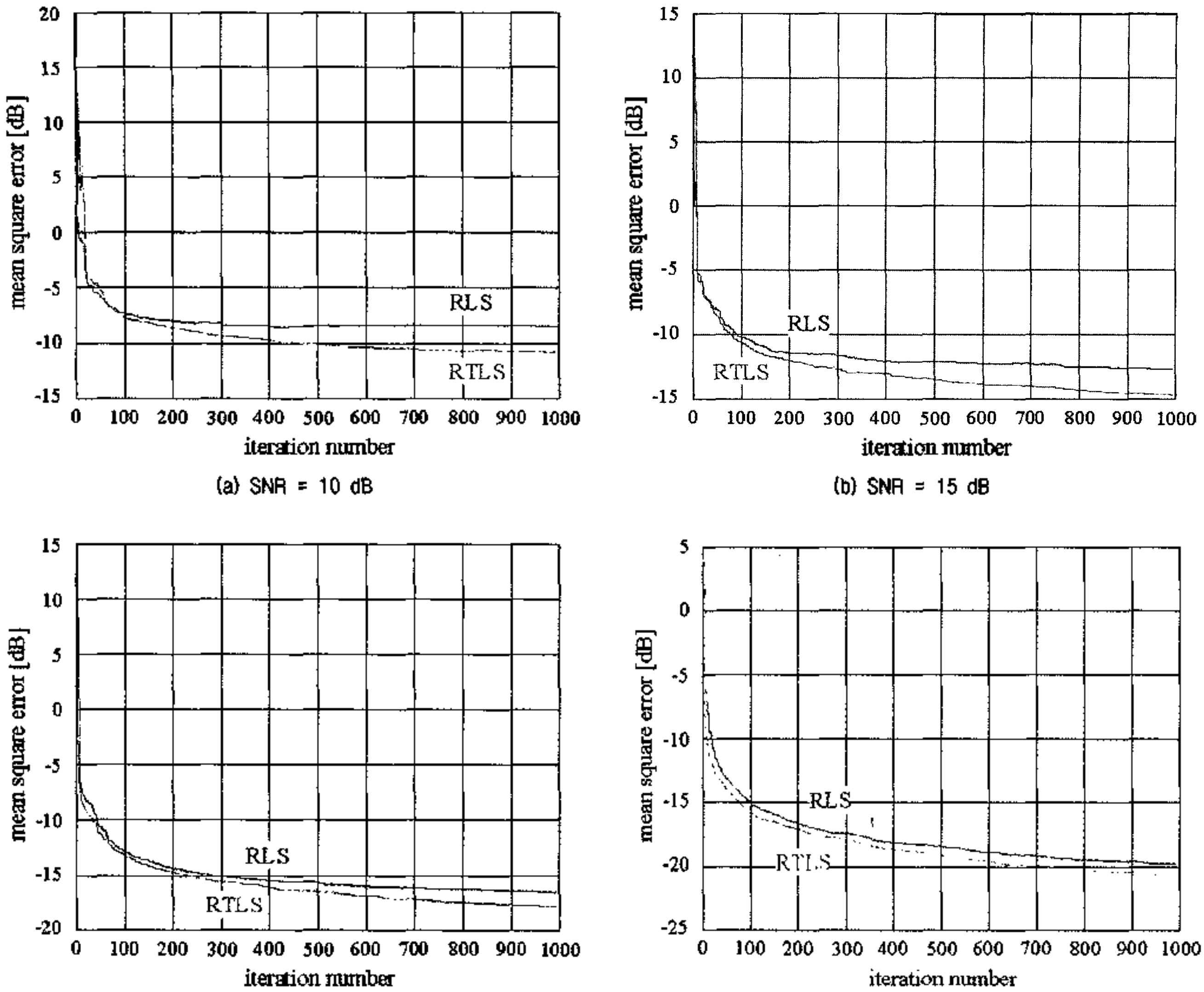
In this case, we can apply the new RTLS algorithm efficiently to the adaptive system identification problem.

IV. Experimental Results

The new RTLS algorithm was compared with the conventional RLS algorithm in an adaptive system identification experiment. In this experiment, the unknown system had a finite impulse response.

$$\mathbf{x}^*=[-0.3\ \ -0.9\ \ 0.8\ \ -0.7\ \ 0.6]^T \tag{19}$$

This system was driven by a binary pseudo-random sequence $\mathbf{a}^*(n)$ which has the values of ±1. The input



(a) SNR = 10 dB (b) SNR = 15 dB

(c) SNR = 20 dB (d) SNR = 25 dB

Figure 2. Comparison of RLS and RTLS algorithms in an adaptive system identification experiment.

and output noise also consisted of white Gaussian noise sequences with various SNRs. The mean squared error

$$\left\| \mathbf{x}(n) - \mathbf{x}^{*} \right\|^{2} \tag{20}$$

where $\mathbf{x}(n)$ is the impulse response estimate was computed from n=1 to 1000 and subsequently averaged over 100 independent experiments. Fig. 2 shows the mean squared error derived from RLS and RTLS with various SNRs. It is observed that the presented RTLS outperforms conventional RLS and provides better performance over a wide range of SNRs, particularly, in low SNR.

V. Conclusions

In this paper, an efficient recursive total least squares algorithm is presented. This algorithm was found to outperform the RLS algorithm. It involves approximately the same computational complexity as RLS algorithm. Further, in order to validate its performance, we applied this algorithm to the adaptive system identification problem in the various noise conditions. In each case, the result showed to be better than that of the RLS.

References

1. S. Haykin, *Adaptive Filter Theory*, 4[th] Ed, Prentice-Hall, 2002.
2. G. H. Golub and C. F. Van Loan, "An analysis of the total least squares problem," *SIAM J. Numer. Anal.*, **17** (6), 883-893, Dec. 1980.
3. K. Pearson, "On lines and planes of closest fit to points in space," *Phil. Mag.*, 2, 559-572, 1901.
4. A. Cichocki and S. I. Amari, *Adaptive Blind Signal and Image Processing*, John Wiley & Sons, 2002.
5. C. E. Davila, "An efficient recursive total least squares algorithm for FIR adaptive filtering," *IEEE Transactions on Signal Processing*, **42** (2), 268-280, 1994.
6. A. Cichocki, R. Unbehauen, "Simplified neural networks for solving linear least squares and total least squares problems in real time," *IEEE Transactions on Neural Networks*, 5 (6), 910-923, 1994.
7. K. Gao, M. O. Ahmad, and M. N. S. Swamy, "A constrained anti-Hebbian learning algorithm for total least-squares estimation with applications to adaptive FIR and IIR filtering," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, **41** (11), 718-729, 1994.
8. W. Chen, M. Chen and J. Zhou, "Adaptively regularized constrained total least-squares image restoration," *IEEE Transactions on Image Processing*, 9 (4), 588-596, 2000.

[Profile]

• Nakjin Choi

Nakjin Choi was born in Pusan, Korea, in 1978. He received the B. E. degree from Daejin University, Korea in 2000 and the M. S. degree from Seoul National University, Korea in 2002.
He started his PH. D. program on acoustical signal processing for underwater communication systems at Seoul National University, Seoul, Korea in 2002.
And he has been working as a Research Engineer at Underwater Acoustics Research Center in Korea from 2000.
His research interests include adaptive filters, array processing and, underwater acoustic and MIMO radio channels.

• Jun-Seok Lim

Jun-Seok Lim was born in Korea in 1964. He received the B. E., the M. S. and PH. D. degrees from Seoul National University, Korea, in 1986, 1988 and 1996 respectively.
He worked as a Research Engineer for Agency for Defense Development in Korea from 1988 to 1993. From 1996 to 1997, he was with LG Electronics Company as a Senior Research Engineer.
Since 1998, he has been with Sejong University in Seoul, Korea as an Assistant Professor. His research interests are in the area of signal processing with communication and array processing in radar and sonar.

• Joon-Il Song

Joon-il Song was born in Korea in 1973. He received the B. E., the M. S., and PH. D. degrees from Seoul National University, Korea, in 1996, 1998 and 2003 respectively. He worked as a Research Engineer for Underwater Acoustics Research Center in Korea from 1997 to 2002. Since 2003, he has been with Samsung Electronics Co. Ltd. in Korea as a Senior Engineer.

• Koeng-Mo Sung

Koeng-Mo Sung was born in Incheon, Korea in 1947. He was in the Department of Electronics Engineering at Seoul National University from 1965 to 1971. He received the Dipl.-Ing. in communication engineering in 1977 and the Dr.-Ing. degree in acoustics in 1982 from Technische Hochschule Aachen, Aachen, Germany. He was a research engineer at RWTH Aachen from 1977 to 1983. Since 1983 he has been with Seoul National University, where he is a Professor in the School of Electrical Engineering. He is a member of the Board of Directors of the Acoustical Society of Korea. His research interests are ultrasonics, musical acoustics and audio signal processing.

APPENDIX. A

Relationship between $\mathbf{v}_{N+1}(n)$ and P(*n*)

We define the correlation matrix **R**(*n*) as follows.

$$
\begin{aligned}
\mathbf{R}(n) &= \overline{\mathbf{A}}^{H}(n)\overline{\mathbf{A}}(n) \\
&= (\mathbf{U}(n)\,\Sigma(n)\,\mathbf{V}^{H}(n))^{H}(\mathbf{U}(n)\,\Sigma(n)\,\mathbf{V}^{H}(n)) \\
&= \mathbf{V}(n)\,\Sigma^{H}(n)\,(\mathbf{U}^{H}(n)\mathbf{U}(n))\,\Sigma(n)\,\mathbf{V}^{H}(n) \qquad (A.1)\\
&= \mathbf{V}(n)\,\Sigma^{H}(n)\,\Sigma(n)\,\mathbf{V}^{H}(n) \\
&= \mathbf{V}(n)\,\Sigma_{R}(n)\,\mathbf{V}^{H}(n) \\
&= \begin{bmatrix} | & & | \\ \mathbf{v}_1(n) & \cdots & \mathbf{v}_{N+1}(n) \\ | & & | \end{bmatrix}
\begin{bmatrix} \sigma_1^2(n) & 0 & \cdots & 0 \\ 0 & \sigma_2^2(n) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{N+1}^2(n) \end{bmatrix}
\begin{bmatrix} | & & | \\ \mathbf{v}_1(n) & \cdots & \mathbf{v}_{N+1}(n) \\ | & & | \end{bmatrix}^{H}
\end{aligned}
$$

Then P(*n*), which is the inverse of matrix **R**(*n*), can be written as

$$
\begin{aligned}
\mathbf{P}(n) &= \mathbf{R}^{-1}(n) \\
&= (\mathbf{V}(n)\Sigma_{R}(n)\mathbf{V}^{H}(n))^{-1} \\
&= (\mathbf{V}^{H}(n))^{-1}(\Sigma_{R}(n))^{-1}(\mathbf{V}(n))^{-1} \qquad (A.2)\\
&= (\mathbf{V}^{-1}(n))^{-1}(\Sigma_{R}(n))^{-1}(\mathbf{V}^{-1}(n)) \\
&= \mathbf{V}(n)\Sigma_{R}^{-1}(n)\mathbf{V}^{H}(n) \\
&= \begin{bmatrix} | & & | \\ \mathbf{v}_1(n) & \cdots & \mathbf{v}_{N+1}(n) \\ | & & | \end{bmatrix}
\begin{bmatrix} \frac{1}{\sigma_1^2(n)} & 0 & \cdots & 0 \\ 0 & \frac{1}{\sigma_2^2(n)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{\sigma_{N+1}^2(n)} \end{bmatrix}
\begin{bmatrix} | & & | \\ \mathbf{v}_1(n) & \cdots & \mathbf{v}_{N+1}(n) \\ | & & | \end{bmatrix}^{H}
\end{aligned}
$$

From Eq.(A.2), we can obtain $\mathbf{v}_{N+1}(n)$ from **P**(*n*).

APPENDIX. B

Summary of Conventional RLS Algorithm

Initialize this algorithm by setting $\mathbf{P}(0)=\delta^{-1}\mathbf{I}$, $\hat{\mathbf{x}}_{LS}(0)=\mathbf{0}$ where δ is a small positive constant, $\mathbf{I}$ is an N×N identity matrix and $\mathbf{0}$ is an N×1 0's vector.	
Compute for each instant of time, n=1,2, ...,M	MAD's
1. $\mathbf{k}(n)=\dfrac{\lambda^{-1}\mathbf{P}(n-1)\mathbf{a}(n)}{1+\lambda^{-1}\mathbf{a}^{H}(n)\mathbf{P}(n-1)\mathbf{a}(n)}$	1. N^2+2N+1
2. $\xi(n)=b(n)-\hat{\mathbf{x}}_{LS}^{H}(n-1)\mathbf{a}(n)$	2. N
3. $\hat{\mathbf{x}}_{LS}(n)=\hat{\mathbf{x}}_{LS}(n-1)+\mathbf{k}(n)\xi^{*}(n)$	3. N
4. $\mathbf{P}(n)=\lambda^{-1}\mathbf{P}(n-1)-\lambda^{-1}\mathbf{k}(n)\mathbf{a}^{H}(n)\mathbf{P}(n-1)$	4. $3N^2$
Total number of MAD's	$4N^2+4N+1$

MAD's stands for number of multiplies, divides, and square roots.