THE JOURNAL OF THE ACOUSTICAL SOCIETY OF KOREA VOL.26, NO.2E 2007. 6 pp. 63~68

A Channel Equalization Algorithm Using Neural Network Based Data Least Squares

Jun-Seok Lim*, Yong-Kuk Pyeon**
*Department of Electronics Engineering Sejong University **Gangwon Provincial College
(Received May 21 2007; Accepted July 4 2007)

Abstract

Using the neural network model for oriented principal component analysis (OPCA), we propose a solution to the data least squares (DLS) problem, in which the error is assumed to lie in the data matrix only. In this paper, we applied this neural network model to channel equalization. Simulations show that the neural network based DLS outperforms ordinary least squares in channel equalization problems.

Keywords: *Data Least Square Method, Oriented Principal Component Analysis, Equalization.*

I. Introduction

Linear least squares (LS) problems involve finding "good" approximate solutions to a set of independent, but inconsistent, linear equations

$$Ax = b \qquad (1)$$

where A is an m x n complex data matrix; b is a complex m × 1 observation vector; and x is a complex n × 1 prediction vector, which is optimally chosen to minimize some kind of squared error measure. It is usually assumed that the underlying noiseless data satisfy (1) with equality. Different classes of LS problems can be defined in terms of the type of perturbation necessary to achieve equality in the system of equations described by (1). For example, in the ordinary least squares (OLS) problem, the error (or perturbation) is assumed to lie in b.

$$Ax_{OLS} = (b+r) \qquad (2)$$

where r is the residual error vector that corresponds to a perturbation in b. The OLS solution vector x_{OLS} is chosen so that the Euclidean (or Frobenius) norm of r is minimized. It is implicitly assumed in the OLS problem that A is completely errorless, and therefore the columns of A are not perturbed in the solution [1]. On the other hand, the total least squares (TLS) problem assumes error in both A and b.

$$(A+E)x_{TLS} = (b+r) \qquad (3)$$

The TLS solution vector is chosen so that the Euclidean norm of [E r] is minimal. Another interesting case that is described and solved in this correspondence assumes that errors occur in A but not b. We call this case the data least squares (DLS) problem because the error is assumed to lie in the data matrix A as indicated by

$$(A+E)x_{DLS} = b \qquad (4)$$

DeGroat, et. al. in [2] developed a close form solution to (4) and demonstrated that it outperformed OLS and TLS in case of noisy data matrix. However, the solution was a kind of batch type algorithm.

In this paper, we propose a neural network model for DLS solution with a neural network model for oriented

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

principal component analysis (OPCA). We applied this neural network model to channel equalization. Simulations for the performance comparison show that the proposed DLS network outperforms ordinary least squares in symbol error rate (SER).

II. Generalized Total Least Squares Problem

Given an unknown system with finite impulse response and assuming that both the input and output are corrupted by the Gaussian white noise, the system should be estimated from the noisy observation of the input and output, as Fig.1. The unknown system is described by

$$\mathbf{h}= [h_0,h_1,\cdots,h_{N-1}]^H \in \mathrm{C}^{N+1} \tag{5}$$

where h may be time-varying or time-invariant. The output is given by

$$d(n) = \mathbf{x}^H(n)\mathbf{h} + n_o(n) \tag{6}$$

where the output noise $n_o(n)$ is a Gaussian white noise with variance σ_o^2 and independent of the input signal, and the noise free input vector is represented as

$$\mathbf{x}(n) = [x(n), x(n-1), \cdots, x(n-N+1)]^T \tag{7}$$

The noisy input vector of the system is given by

$$\tilde{\mathbf{x}}(n) = \mathbf{x}(n) + \mathbf{n}_i(n) \in C^{N\times 1} \tag{8}$$

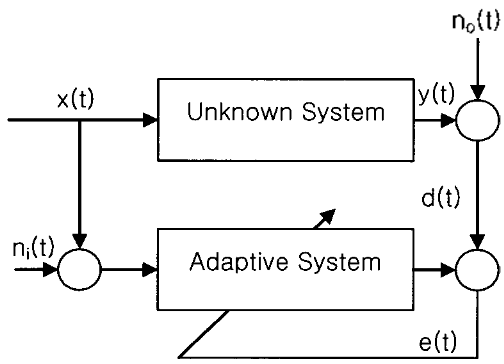

Fig. 1. The model of generalized total least square.

where $\mathbf{n}_i(n) = [n_i(n), n_i(n-1), \cdots, n_i(n-N+1)]^T$ and the input noise $n_i(n)$ is the Gaussian white noise with variance σ_i^2.

Notice that the input noise may originate from the measured error, interference, quantized noise and so on. Hence, we adopt a more general signal model than the least squares based estimation. Moreover, the augmented data vector is defined as

$$\bar{\mathbf{x}}(n) = \left[\tilde{\mathbf{x}}^T(n), d(n)\right]^T \in \mathrm{C}^{(N+1)\times 1} \tag{9}$$

The correlation matrix of the augmented data vector has the following structure

$$\bar{\mathbf{R}} = \begin{bmatrix} \tilde{\mathbf{R}} & \mathbf{p} \\ \mathbf{p}^H & c \end{bmatrix} \tag{10}$$

where $\mathbf{p} = E\{\tilde{\mathbf{x}}(n)d^*(n)\}$ and $c = E\{d(n)d^*(n)\}$, $\tilde{\mathbf{R}} = E\{\tilde{\mathbf{x}}(n)\tilde{\mathbf{x}}^H(n)\} = \mathbf{R} + \sigma_i^2\mathbf{I}$, $\mathbf{R} = E\{\mathbf{x}(n)\mathbf{x}^H(n)\}$. We can further establish that $\mathbf{p} = \mathbf{R}^H\mathbf{h}$ and $c = \mathbf{h}^H\tilde{\mathbf{R}}\mathbf{h} + \sigma_o^2$.

Define the constrained Rayleigh quotient as

$$J(\mathbf{w}) = \frac{[\mathbf{w}^T,-1]\bar{\mathbf{R}}[\mathbf{w}^T,-1]^H}{[\mathbf{w}^T,-1]\bar{\mathbf{D}}[\mathbf{w}^T,-1]^H} \tag{11}$$

where $\bar{\mathbf{D}} = \begin{bmatrix} \mathbf{I} & 0 \\ 0 & \gamma \end{bmatrix}$ with $\gamma = \frac{\sigma_o^2}{\sigma_i^2}$ [3]. The generalized total least square solution is obtained by solving

$$\min_{\mathbf{w}} J(\mathbf{w}) \tag{12}$$

DLS is a special case in (11) with =0 [3].

III. Oriented PCA and Its Neural Network Model

3.1. Oriented PCA (OPCA)

In this section we extend the standard principal component analysis problem by introducing OPCA [4,5] which corresponds to the generalized eigenvalue problem of two random signals and bears the same relationship to generalized eigenvalue decomposition (GED) as PCA bears to ordinary eigenvalue decomposition (ED). More precisely, the goal is to find the direction vector w that maximizes the signal−to−signal ratio

$$J_{OPC} = \frac{E\{(\tilde{\mathbf{w}}^H \mathbf{x}_1)^2\}}{E\{(\tilde{\mathbf{w}}^H \mathbf{x}_2)^2\}} = \frac{\tilde{\mathbf{w}}^H \mathbf{R}_1 \tilde{\mathbf{w}}}{\tilde{\mathbf{w}}^H \mathbf{R}_2 \tilde{\mathbf{w}}} \tag{13}$$

where $\mathbf{R}_1 = E\{\mathbf{x}_1\mathbf{x}_1^H\}$ and $\mathbf{R}_2 = E\{\mathbf{x}_2\mathbf{x}_2^H\}$. We assume that $\mathbf{R}_2$ is strictly positive definite, hence nonsingular. Quite often $\{x_{1k}\}$ and $\{x_{2k}\}$are stationary stochastic processes, whence $\mathbf{R}_1 = E\{\mathbf{x}_{1k}\mathbf{x}_{1k}^H\}$ and $\mathbf{R}_2 = E\{\mathbf{x}_{2k}\mathbf{x}_{2k}^H\}$ and OPCA is still defined via (13). As usual there is little difference between random vectors and stationary random processes, and we'll use the term OPCA for both cases interchangeably.

The optimal solution to (13) will be called the principal oriented component of the pair (x_1, x_2). Referring to Fig. 2, the adjective "oriented" is justified by the fact that the principal component of x_1 is now steered by the distribution of x_2: it will be oriented toward the directions where v has minimum energy while trying to maximize the projection energy of x_1. J_{opc} is nothing but the generalized Rayleigh quotient for the matrix pencil $(\mathbf{R}_1, \mathbf{R}_2)$, so the principal oriented component is the principal generalized eigenvector of the symmetric generalized eigenvalue problem [4,5].

$$\mathbf{R}_1 \tilde{\mathbf{w}} = \lambda \mathbf{R}_2 \tilde{\mathbf{w}} \tag{14}$$

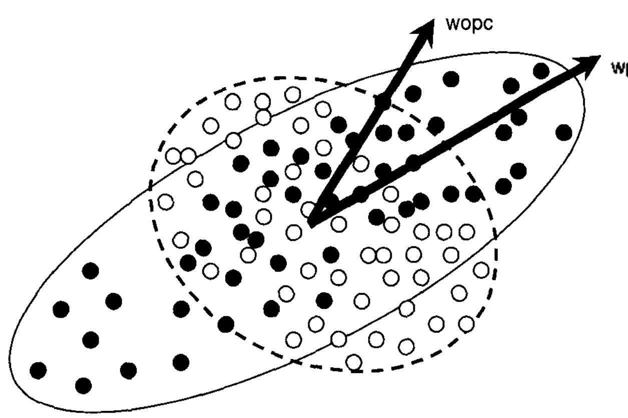

Fig. 2. A visual interpretation of oriented principal component analysis: although the principal component Wpc is along the major axis of the signal, the oriented principal component Wopc is steered by the distribution of the noise.

3.2. Network Models for OPC Extraction

If we initially focus on the extraction of the first component in [4,5], the maximum value of J_{opc} in (13) is the principal generalized eigenvalue λ_1. Therefore, the function

$$V(\tilde{\mathbf{w}}) = \frac{1}{2}(\lambda_1 - J_{OPC}(\tilde{\mathbf{w}})) \tag{15}$$

is such that $V(\tilde{\mathbf{w}}) > 0$, and $V(\tilde{\mathbf{w}}) = 0$ only for $\tilde{\mathbf{w}} = \mathbf{e}_1$, so V may serve as a Lyapunov energy function for a system to be proposed. The proper gradient descent algorithm would be

$$\frac{d\tilde{\mathbf{w}}}{dt} = -\nabla V = \frac{1}{\tilde{\mathbf{w}}^H \mathbf{R}_2 \tilde{\mathbf{w}}}\left(\mathbf{R}_1 \tilde{\mathbf{w}} - \frac{\tilde{\mathbf{w}}^H \mathbf{R}_1 \tilde{\mathbf{w}}}{\tilde{\mathbf{w}}^H \mathbf{R}_2 \tilde{\mathbf{w}}} \mathbf{R}_2 \tilde{\mathbf{w}}\right) \tag{16}$$

with the globally asymptotically stable fixed point $\tilde{\mathbf{w}} = \mathbf{e}_1$.

In fact, even the simpler equation

$$\frac{d\tilde{\mathbf{w}}}{dt} = \left(\mathbf{R}_1 \tilde{\mathbf{w}} - \frac{\tilde{\mathbf{w}}^H \mathbf{R}_1 \tilde{\mathbf{w}}}{\tilde{\mathbf{w}}^H \mathbf{R}_2 \tilde{\mathbf{w}}} \mathbf{R}_2 \tilde{\mathbf{w}}\right) \tag{17}$$

is stable since

$$\frac{dV}{dt} = \frac{d\tilde{\mathbf{w}}^H}{dt} \nabla V = -\frac{1}{\tilde{\mathbf{w}}^H \mathbf{R}_2 \tilde{\mathbf{w}}}\left\| \mathbf{R}_1 \tilde{\mathbf{w}} - \frac{\tilde{\mathbf{w}}^H \mathbf{R}_1 \tilde{\mathbf{w}}}{\tilde{\mathbf{w}}^H \mathbf{R}_2 \tilde{\mathbf{w}}} \mathbf{R}_2 \tilde{\mathbf{w}} \right\|^2 \le 0 \tag{18}$$

and again the point $\tilde{\mathbf{w}} = \mathbf{e}_1$ is the globally asymptotically stable attractor.

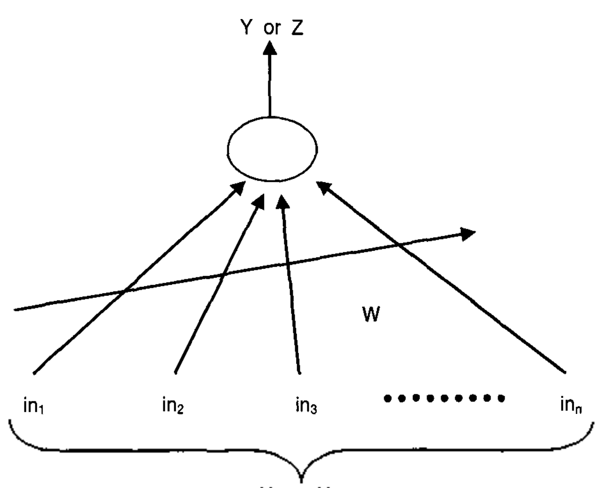

Fig. 3. OPC linear neuron unit.

IV. Neural Network-based Data Least Squares Algorithm

We can apply the generalized eigendecomposition method in section III to solution of DLS. If we modify (11) and (12), the object function for DLS becomes as follows.

$$\tilde{J}(\mathbf{w}) = \frac{\tilde{\mathbf{w}}^H \overline{\mathbf{D}} \tilde{\mathbf{w}}}{\tilde{\mathbf{w}}^H \overline{\mathbf{R}} \tilde{\mathbf{w}}} = \frac{[\mathbf{w}^H,-1]\overline{\mathbf{D}}[\mathbf{w}^T,-1]^T}{[\mathbf{w}^H,-1]\overline{\mathbf{R}}[\mathbf{w}^T,-1]^T} \qquad (19)$$

The DLS solution can be derive as (20). We apply the recursive algorithm in section III for the maximization of (20).

$$\max_{\tilde{\mathbf{w}}} \tilde{J}(\tilde{\mathbf{w}}), \text{ and then } \mathbf{w} = \tilde{\mathbf{w}}(1:N)/(-\tilde{\mathbf{w}}(N+1)) \qquad (20)$$

where $\tilde{\mathbf{w}}(1:N)$ is a vector with the elements from the *1-st* to the *N-th*, and $\tilde{\mathbf{w}}(N+1)$ is the $(N+1)$-th element in $\tilde{\mathbf{w}}$. When we apply the OPCA to (19), we have a update equation.

$$\Delta\tilde{\mathbf{w}} = \frac{1}{(\tilde{\mathbf{w}}^H \overline{\mathbf{R}} \tilde{\mathbf{w}})^2}\left((\tilde{\mathbf{w}}^H \overline{\mathbf{R}} \tilde{\mathbf{w}})\overline{\mathbf{D}}\tilde{\mathbf{w}} - (\tilde{\mathbf{w}}^H \overline{\mathbf{D}} \tilde{\mathbf{w}})\overline{\mathbf{R}}\tilde{\mathbf{w}}\right) \qquad (21)$$

$$\begin{aligned}\tilde{\mathbf{w}}(n) = {} & \tilde{\mathbf{w}}(n-1) \\ & \beta(\tilde{\mathbf{w}}^H(n-1)\overline{\mathbf{R}}(n)\tilde{\mathbf{w}}(n-1))\overline{\mathbf{D}}\tilde{\mathbf{w}}(n-1) \\ & -\beta(\tilde{\mathbf{w}}^H(n-1)\tilde{\mathbf{w}}(n-1))\overline{\mathbf{R}}(n)\tilde{\mathbf{w}}(n-1)\end{aligned} \qquad (22)$$

where

$\overline{\mathbf{R}}(n) = \lambda_f \overline{\mathbf{R}}(n-1) + \overline{\mathbf{x}}(n)\overline{\mathbf{x}}^H(n)$ and λ_f is a forgetting factor.

We summarize the algorithm in table 1.

Table I. OPCA BASED Data LeastSquare (NN-DLS) Algorithm.

1. Initialize λ_f, β, $\overline{\mathbf{x}}(0) = [\mathbf{x}^T(0), d(0)]$, $\tilde{\mathbf{w}}(0) = [\mathbf{w}^T(0), -1]$ with the w(0) $\in C^{N\times 1}$ to a random vector
2. Fill the matrix Q(0) $\in C^{N\times N}$ with small random values For j 〉 0
3. Compute $z(j) = \tilde{\mathbf{w}}^H(j-1)\overline{\mathbf{x}}(j)$
4. Update $\overline{\mathbf{R}}$ as $\overline{\mathbf{R}}(j) = \lambda_f \overline{\mathbf{R}}(j-1) + \overline{\mathbf{x}}(j)\overline{\mathbf{x}}^H(j)$
5. Compute $\mathbf{z}_1(j) = \overline{\mathbf{R}}(j)\tilde{\mathbf{w}}(j-1)$ and $z_2(j) = \tilde{\mathbf{w}}^H(j-1)\mathbf{z}_1(j)$
6. Update the weight vector as $\tilde{\mathbf{w}}(j) = \tilde{\mathbf{w}}(j-1) + \beta\left(z_2(j)\overline{\mathbf{D}}\tilde{\mathbf{w}}(j-1) - (\mathbf{w}^H(j-1)\mathbf{w}(j-1))\mathbf{z}_1(j)\right)$
7. Normalize the weight vector
8. $\mathbf{w}(j) = \tilde{\mathbf{w}}(1:n-1)/(-\tilde{\mathbf{w}}(n+1))$ loop

V. A Channel Equalization Application

In this section, we demonstrate the usefulness of the DLS method by comparing it with the optimal method and OLS methods when applied to a channel equalization problem. The channel equalization problem is graphically described by the block diagram in Fig. 4. Basically, the solution vector, $\mathbf{w} = [w_1, w_2, \cdots, w_p]^T$ represents an FIR approximant inverse filter to the channel characteristic *H(z)*. The output of the inverse (equalization) filter can be written in matrix form using the output of the channel as input to the finite impulse response (FIR) equalization filter. The output of the equalized channel should be approximately equal to the original input

$$\begin{bmatrix}\tilde{s}_{p-1}\\ \tilde{s}_p\\ \vdots\\ \tilde{s}_{N-1}\end{bmatrix} = \begin{bmatrix} v_{p-1} & \cdots & v_1 & v_0\\ v_p & \cdots & v_2 & v_1\\ \vdots & \vdots & \vdots & \vdots\\ v_{N-1} & \cdots & v_{N-p+1} & v_{N-p}\end{bmatrix}\begin{bmatrix}w_1\\ w_2\\ \vdots\\ w_p\end{bmatrix} \approx \begin{bmatrix}s_{p-1}\\ s_p\\ \vdots\\ s_{N-1}\end{bmatrix} \qquad (23)$$

where p is the FIR filter order; and N is the total number of output samples. In this problem, we assume that the left side in (23) is known without error because the input training signal is assumed to be known without error. It is easy to see that (23) has the form of (4).

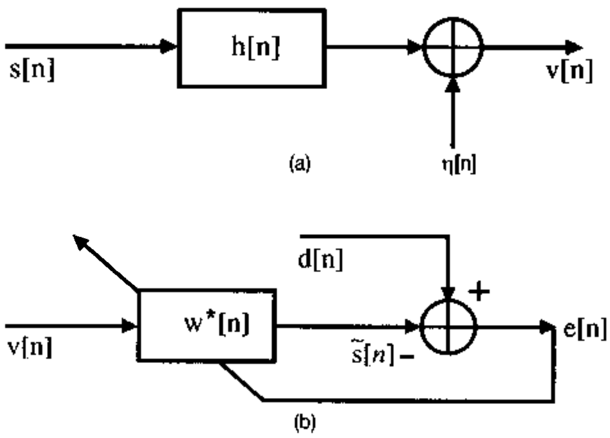

Fig. 4. Transmission and Equalization model: (a) received signal model, (b) equalizer model (s[n]:transmitted signal, h[n]: channel model, η[n]: additive noise, v[n]:received signal, d[n]:training signal).

For the simulation, a well-known complex nonminimum-phase channel model introduced by Cha and Kassam [6] is used to evaluate the proposed neural network based data least square (NN-DLS) equalizer performance for 4-PAM signaling. Although the length of the channel is short, the channel model cannot only simulate the phase change from boundary reflection but also do the nonminimum phase characteristics of the channel in the room acoustics or in the underwater communication. The channel output $v(n)$ (which is also the input of the equalizer) is given by

$$v(n)=(0.34-j0.27)s(n)+(0.87+j0.43)s(n-1)+(0.34-j0.21)s(n-2)+\eta(n),\quad \eta(n)\sim N(0,0.01) \tag{24}$$

where N(0, 0.01) means the white Gaussian noise (of the nonminimum-phase channel) with mean 0 and variance 0.01. 4-PAM symbol sequence $s(n)$ is passed through the channel and the sequence $s(n)$ are valued from the set $\{\pm1, \pm3\}$. All the equalizers, the least mean square (LMS) based equalizer and the proposed NN-DLS based equalizer, were trained with 1000 data symbols at 15 dB SNR. The LMS is an simple adaptive algorithm for the OLS problem. The order of equalizer is set to 7.

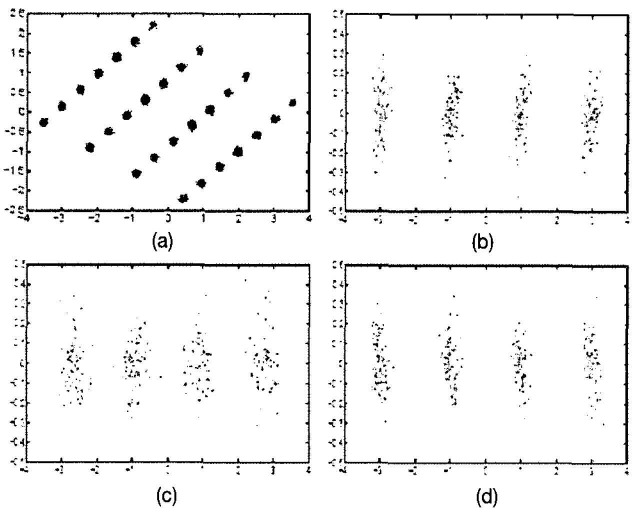

Fig. 5. Performance comparison of three equalizers: (a) scatter diagram of received signals, (b) scatter diagram of optimal (Wiener Solution) equalizer, (c) scatter diagram of LMS equalizer, (d) scatter diagram of the proposed equalizer.

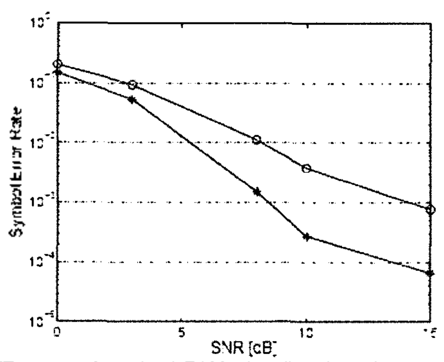

Fig. 6. SER comparison in 4-PAM signaling (-*-: the proposed algorithm, -o-: LMS algorithm).

Fig. 5 (a) shows the distribution of the input data of the different equalizers. This figure shows received signals scattered severely due to transmission channel effect. Fig. 5 (b), (c) and (d) show the scatter diagrams of the outputs of the three equalizers, optimal (Wiener solution), LMS based and the NN-DLS based, respectively. As observed from Fig. 5, the equalized signal by the proposed algorithm centres on $\{\pm1, \pm3\}$and it is almost the same as the equalized signals by the optimal equalizer which is derived from the Wiener solution. It leads the conclusion that the proposed NN-DLS outperforms the LMS algorithm. Moreover, it estimates almost the same as optimal equalizer.

For the performance comparison, we show the symbol error rate (SER) for the proposed equalizer and an LMS-based equalizer. They were trained in several SNRs, from 0 dB to 20 dB. We set the step-size to 10^{-4} for both equalizers. Fig. 6shows the SER in the above linear nonminmum phase channel with 4-PAM sequences. It shows that the proposed algorithm outperforms the LMS-based equalizer in the entire SNR range. Therefore, the proposed DLS algorithm outperforms the OLS algorithm.

VI. Conclusion

In this paper, we proposed a recursive algorithm for data least square (DLS) solution. Channel equalization simulations were performed to compare the proposed algorithm with the algorithms in OLS and we found better performance over OLS methods.

Acknowledgment

This result was supported by ADD in Korea.

References

1. G. H. Golub and C. F. Van Loan, "An analysis of the total least squares problem," SIAMJ. Num. Anal., 17, 883- 893, 1980.
2. R.D. DeGroat and E. M. Dowling, "The Data Least Squares and Channel Equalization," IEEE Trans. Signal Processing, **41**, 407~411, 1993.
3. C.E. Davila, "Line Search Algorithm for Adaptive Filtering," IEEE Trans. Signal Processing, 42, 2490~2494, 1994.
4. K.I. Diamantaras and S.Y. Kung, *Principal Component Neural Networks: Theory and Applications,* (Wiley, 1996), pp. 186~202.
5. E.F. Deprettere, editor, *SVD and Signal Processing,* (Elsevier Science Publishers, New York, 1973), pp. 209~232.
6. Cha, I. and S.A. Kassam, "Channel equalization using adaptive complex radial basis function networks," IEEE J. Sel. Area. Comm., 13, 122-131, 1995.

[Profile]

• **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 Associate Professor. His research interests are in the area of adaptive signal processing.

• **Yong-Kuk Pyeon**

Yong-Kuk Pyeon was born in Korea in 1966. He received Ph.D. degrees from Sejong University, Korea, in 2004.
Since 2004, he has been with Gangwon Provincial College in Gangwon-Do, Korea as an Assistant Professor. His research interests are in the area of signal processing.