

A Matrix Pseudo-Inversion Lemma and Its Application to Block-Based Adaptive Blind Deconvolution for MIMO Systems

Kiyotaka Kohno*, Yujiro Inouye† and Mitsuru Kawamoto‡

*†Department of Electronic and Control Systems Engineering, Shimane University,
1060 Nishikawatsu, Matsue, Shimane 690-8504, Japan

†Laboratory for Advanced Brain Signal Processing, Brain Science Institute, RIKEN,
2-1 Hirosawa, Wako, Saitama 351-0198, Japan

‡Information Technology Research Institute,
National Institute of Advanced Industrial Science and Technology,
Central 2, 1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan

*kohno@yonago-k.ac.jp, †inouye@riko.shimane-u.ac.jp, ‡m.kawamoto@aist.go.jp

Abstract—The matrix inversion lemma gives an explicit formula of the inverse of a positive-definite matrix A added to a block of dyads (represented as BB^H) as follows:

$$(A + BB^H)^{-1} = A^{-1} - A^{-1}B(I + B^H A^{-1}B)^{-1}B^H A^{-1}.$$

It is well-known in the literature that this formula is very useful to develop a block-based recursive least-squares algorithm for the block-based recursive identification of linear systems or the design of adaptive filters.

We extend this result to the case when the matrix A is singular, and present a matrix pseudo-inversion lemma. Based on this result, we propose a block-based adaptive multicahnnel super-exponential algorithm (BAMSEA). We present simulation results for the performance of the block-based algorithm in order to show the usefulness of the matrix pseudo-inversion lemma.

I. INTRODUCTION

The familiar matrix inversion lemma states that the inverse of a positive-definite $n \times n$ matrix A added to a block of dyads (represented as BB^H) can be represented as

$$(A + BB^H)^{-1} = A^{-1} - A^{-1}B(I + B^H A^{-1}B)^{-1}B^H A^{-1}, \quad (1)$$

where B is an $n \times r$ matrix and the superscript H denotes the complex conjugate transpose (or Hermitian) operation. It is well-known in the literature that this formula is very useful to develop a block-based recursive least-squares algorithm for the block-based recursive identification [1], [2] or the design of adaptive filters [3].

In the late 1980s, Ogawa extended the matrix inversion lemma in (1) to the case when A is positive semidefinite [4]. However, his extension is valid under the condition that the range of A includes the range of B , but this condition is not satisfied for adaptive signal processing in non-stationary environments.

In the present paper, we extend the matrix inversion lemma in (1) to the case when the matrix A is positive semidefinite without the above condition for the ranges of the relevant matrices, and present a matrix pseudo-inversion lemma. Such a singular case may occur in a situation where a problem dealt with is overdetermined in the sense that it has more equations than unknowns. In particular, we encountered this singular situation when we developed a sample-based adaptive version of the super-exponential method (SEM) for the blind deconvolution of multi-input multi-output (MIMO) systems, where the number of its outputs is greater than the number of its inputs. It should be noted that our previous work on the matrix pseudo-inversion lemma is restricted to the case when the added matrix BB^H is a single dyad (i.e., B is a column vector) [9], [10].

After the presentation of the matrix pseudo-inversion lemma, we apply this lemma to block-based adaptive blind deconvolution of a MIMO system and we propose a block-based adaptive version of the multichannel super-exponential algorithm for the blind deconvolution.

We also include simulation results for the performance of the proposed algorithm in order to show its effectiveness, where we compare the performance of the proposed algorithm using the lemma

with that of the algorithm using the built-in function in MATLAB version 7.1.0 for calculating pseudoinverses of the relevant matrices instead of using the lemma.

The present paper uses the following notation: Let Z denote the set of all integers. Let C denote the set of all complex numbers. Let $C^{m \times n}$ denote the set of all $m \times n$ matrices with complex components. The superscripts T , $*$ and \dagger denote, respectively, the transpose, the complex conjugate and the (Moore-Penrose) pseudoinverse operations of a matrix. The symbol \oplus denotes the direct sum of subspaces or the direct sum of matrices and the superscript \perp denotes the orthogonal complement of a subspace [6]. A matrix $D \in C^{n \times r}$ is called a *dyad* (or *dyadic* matrix) if D can be represented as $D = bc^H$ with $b \in C^n$ and $c \in C^r$. Thus a Hermitian dyadic matrix D can be described as $D = bb^H$. The range space of $E \in C^{n \times r}$ is denoted by $R(E)$ [5]. Let $i = 1, \bar{n}$ stand for $i = 1, 2, \dots, n$.

II. MATRIX PSEUDO-INVERSION LEMMA: A GENERAL CASE WITH A BLOCK OF DYADS

The following proposition gives an explicit formula of the pseudo-inverse of a positive semidefinite Hermitian matrix A added to a block of Hermitian dyads (represented as BB^H).

Proposition 1: Let $A \in C^{n \times n}$ be a positive semidefinite Hermitian matrix, and $B \in C^{n \times r}$ be a matrix and decomposed uniquely as

$$B = B_1 \oplus B_2 \quad \text{with} \quad R(B_1) \subset R(A) \quad \text{and} \quad R(B_2) \subset R(A)^\perp. \quad (2)$$

Let Q be defined as

$$Q := A + BB^H \in C^{n \times n}. \quad (3)$$

Then the pseudo-inverse Q^\dagger of the matrix Q is explicitly expressed, depending on the values of matrices B_1 and B_2 , as follows:

1) If $B_2 = 0$, then

$$Q^\dagger = A^\dagger - A^\dagger B_1 (I + B_1^H A^\dagger B_1)^{-1} B_1^H A^\dagger. \quad (4)$$

2) If $B_2 \neq 0$ and $B_1 = 0$, then

$$Q^\dagger = A^\dagger + (B_2^H)^\dagger B_2^\dagger. \quad (5)$$

3) If $B_2 \neq 0$ and $B_1 \neq 0$, then

$$Q^\dagger = Q_B^\dagger - Q_B^\dagger [B_1, B_2] Q_D^{-1} [B_1, B_2]^H Q_B^\dagger, \quad (6)$$

where Q_B^\dagger and Q_D^{-1} are respectively defined by

$$\begin{aligned} Q_B^\dagger &:= (A + B_1 B_1^H + B_2 B_2^H)^\dagger \\ &= A^\dagger - A^\dagger B_1 (I + B_1^H A^\dagger B_1)^{-1} B_1^H A^\dagger + (B_2^H)^\dagger B_2^\dagger, \end{aligned} \quad (7)$$

and

$$\begin{aligned} Q_D^{-1} &:= (P + [B_1, B_2]^H Q_B^\dagger [B_1, B_2])^{-1} \\ &= \left[\begin{array}{c|c} -\Delta^{-1} B_2^H Q_B^\dagger B_2 & \Delta^{-1} \\ \hline I + B_1^H Q_B^\dagger B_1 \Delta^{-1} B_2^H Q_B^\dagger B_2 & -B_1^H Q_B^\dagger B_1 \Delta^{-1} \end{array} \right] \end{aligned} \quad (8)$$

with

$$\mathbf{P} := \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \in \mathbf{R}^{2r \times 2r}, \quad (9)$$

and

$$\mathbf{\Delta} := \mathbf{I} - \mathbf{B}_2^H \mathbf{Q}_B^\dagger \mathbf{B}_2 \mathbf{B}_1^H \mathbf{Q}_B^\dagger \mathbf{B}_1. \quad (10)$$

Here $\mathbf{R}^{2r \times 2r}$ denotes the set of all $2r \times 2r$ matrices with real components.

All the proofs of Proposition, Corollary and Theorem in this paper are omitted for page limit. They will appear in a forthcoming paper.

Remark 1: A technical important fact in Proposition 1 is that there exists really the inverse of the matrix \mathbf{Q}_D defined as

$$\mathbf{Q}_D := \mathbf{P} + [\mathbf{B}_1, \mathbf{B}_2]^H \mathbf{Q}_B^\dagger [\mathbf{B}_1, \mathbf{B}_2] \quad (11)$$

in (8) even if \mathbf{P} is not positive definite. The proof of the existence is not easy and requires a notion of orthogonal projectors along with a geometric approach to linear transformations [5].

It can be seen that the first and the second expressions of the pseudoinverse given in Proposition 1 can be included as special cases in the third expression of the pseudoinverse given in (6). Namely, we have the following corollary.

Corollary 1: Under the same conditions in Proposition 1, it follows that

$$\begin{aligned} \mathbf{Q}^\dagger &= (\mathbf{A} + \mathbf{B}\mathbf{B}^H)^\dagger \\ &= \mathbf{Q}_B^\dagger - \mathbf{Q}_B^\dagger [\mathbf{B}_1, \mathbf{B}_2] \mathbf{Q}_D^{-1} [\mathbf{B}_1, \mathbf{B}_2]^H \mathbf{Q}_B^\dagger, \end{aligned} \quad (12)$$

where \mathbf{Q}_B^\dagger and \mathbf{Q}_D^{-1} are defined by (7) and (8), respectively.

III. BLOCK-BASED ADAPTIVE MULTICHANNEL SUPER-EXPONENTIAL ALGORITHM

Let us consider a MIMO system with n inputs and m outputs as described by

$$\mathbf{y}(t) = \sum_{k=-\infty}^{\infty} \mathbf{H}^{(k)} \mathbf{s}(t-k), \quad t \in \mathbf{Z}, \quad (13)$$

where $\mathbf{y}(t)$ is an m -column vector of output signals, $\mathbf{s}(t)$ is an n -column vector of input (or source) signals, $\mathbf{H}^{(k)}$ is an $m \times n$ matrix sequence called the impulse response. The transfer function of the channel is defined by

$$\mathbf{H}(z) = \sum_{k=-\infty}^{\infty} \mathbf{H}^{(k)} z^{-k}, \quad z \in \mathbf{C}. \quad (14)$$

It is assumed for theoretical analysis that the noise is absent in (13).

To recover the source signals, we process the output signals by an $n \times m$ equalizer (or deconvolver) $\mathbf{W}(z)$ described by

$$\mathbf{z}(t) = \sum_{k=-\infty}^{\infty} \mathbf{W}^{(k)} \mathbf{y}(t-k), \quad t \in \mathbf{Z}. \quad (15)$$

The objective of multichannel blind deconvolution is to construct an equalizer that recovers the original source signals only from the measurements of the corresponding outputs.

We put the following assumptions on the channel and the source signals [8]:

A1) The transfer function $\mathbf{H}(z)$ is stable and has full column rank on the unit circle $|z| = 1$ [this implies that the unknown system has less inputs than outputs, i.e., $n \leq m$, and there exists a left stable inverse of the unknown system].

A2) The input sequence $\{\mathbf{s}(t)\}$ is a complex, zero-mean, non-Gaussian random vector process with element processes $\{s_i(t)\}$, $i = \overline{1, n}$ being mutually independent. Moreover, each element process $\{s_i(t)\}$ is an i.i.d. process with a variance $\sigma_i^2 \neq 0$ and a fourth-order cumulant $\gamma_i \neq 0$. The variances σ_i^2 's and the fourth-order cumulants γ_i 's are unknown.

A3) The equalizer $\mathbf{W}(z)$ is an FIR channel of sufficient length L so that the truncation effect can be ignored.

Under A3, let us consider an FIR equalizer with the transfer function $\mathbf{W}(z)$ given by

$$\mathbf{W}(z) = \sum_{k=L_1}^{L_2} \mathbf{W}^{(k)} z^{-k}, \quad (16)$$

where L_1 and L_2 are respectively the first and the last superscripted numbers of the tap coefficients $\mathbf{W}^{(k)}$'s of the equalizer $\mathbf{W}(z)$, and the length $L := L_2 - L_1 + 1$ is taken to be sufficiently large. Let $\tilde{\mathbf{w}}_i$ be the mL -column vector consisting of the tap coefficients (corresponding to the i th output) of the equalizer defined by

$$\tilde{\mathbf{w}}_i := [\mathbf{w}_{i,1}^T, \mathbf{w}_{i,2}^T, \dots, \mathbf{w}_{i,m}^T]^T \in \mathbf{C}^{mL}, \quad (17)$$

$$\mathbf{w}_{i,j} := [w_{i,j}^{(L_1)}, w_{i,j}^{(L_1+1)}, \dots, w_{i,j}^{(L_2)}]^T \in \mathbf{C}^L, \quad (18)$$

where $w_{i,j}^{(k)}$ is the (i, j) th element of matrix $\mathbf{W}^{(k)}$.

Inouye and Tanebe [8] proposed the *multichannel super-exponential algorithm* (MSEA) for finding the tap coefficient vectors $\tilde{\mathbf{w}}_i$'s of the equalizer $\mathbf{W}(z)$, of which each iteration consists of the following two steps:

$$\tilde{\mathbf{w}}_i^{[1]} = \tilde{\mathbf{R}}^\dagger \tilde{\mathbf{d}}_i \quad \text{for } i = \overline{1, n}, \quad (19)$$

$$\tilde{\mathbf{w}}_i^{[2]} = \frac{\tilde{\mathbf{w}}_i^{[1]}}{\sqrt{\tilde{\mathbf{w}}_i^{[1]H} \tilde{\mathbf{R}} \tilde{\mathbf{w}}_i^{[1]}}} \quad \text{for } i = \overline{1, n}, \quad (20)$$

where $(\cdot)^{[1]}$ and $(\cdot)^{[2]}$ stand respectively for the result of the first step and the result of the second step. Let $\tilde{\mathbf{y}}(t)$ be the mL -column vector consisting of the L consecutive inputs of the equalizer defined by

$$\tilde{\mathbf{y}}(t) := [\tilde{\mathbf{y}}_1(t)^T, \tilde{\mathbf{y}}_2(t)^T, \dots, \tilde{\mathbf{y}}_m(t)^T]^T \in \mathbf{C}^{mL}, \quad (21)$$

$$\tilde{\mathbf{y}}_i(t) := [y_i(t-L_1), y_i(t-L_1-1), \dots, y_i(t-L_2)]^T \in \mathbf{C}^L, \quad (22)$$

where $y_i(t)$ is the i th element of the output vector $\mathbf{y}(t)$ of the channel in (13). Then the correlation matrix $\tilde{\mathbf{R}}$ is represented as

$$\tilde{\mathbf{R}} = E [\tilde{\mathbf{y}}^*(t) \tilde{\mathbf{y}}^T(t)] \in \mathbf{C}^{mL \times mL}, \quad (23)$$

and the fourth-order cumulant vector $\tilde{\mathbf{d}}_i$ is represented as

$$\begin{aligned} \tilde{\mathbf{d}}_i &= \text{cum}(z_i(t), z_i(t), z_i^*(t), \tilde{\mathbf{y}}^*(t)) \\ &= E [z_i(t)^2 z_i^*(t) \tilde{\mathbf{y}}^*(t)] \\ &\quad - 2E [z_i(t)^2] E [z_i(t) \tilde{\mathbf{y}}^*(t)] \\ &\quad - E [z_i^2(t)] E [z_i^*(t) \tilde{\mathbf{y}}^*(t)] \in \mathbf{C}^{mL}, \end{aligned} \quad (24)$$

where $E[x]$ denotes the expectation of a random variable x . We note that the last term can be ignored in case of $E[s_i(t)^2] = 0$, in which case $E[z_i(t)^2] = 0$ for all $i = \overline{1, n}$.

Besides, we obtain (from (32) and (39) in [8])

$$\tilde{\mathbf{d}}_i \in \text{Im } \tilde{\mathbf{R}} \quad \text{for } i = \overline{1, n}. \quad (25)$$

We should note that the stationarity of the input process $\{\mathbf{s}(t)\}$ (or the assumption A2)) ensures the relation (25), and that the relation (25) means that the vector $\tilde{\mathbf{w}}_i^{[1]}$ obtained by (19) satisfies

$$\tilde{\mathbf{R}} \tilde{\mathbf{w}}_i^{[1]} = \tilde{\mathbf{d}}_i \quad (26)$$

Consider the batch algorithm in (19) and (20). The equation (20) constrains a weighted norm of vector $\tilde{\mathbf{w}}_i$ to equal one, and thus we assume this constraint is always satisfied using a normalization or an automatic gain control (AGC) of $\tilde{\mathbf{w}}_i$ at each discrete (or sample) time t . To develop an adaptive version of (19), we must specify the dependency of each time t and rewrite (19) as

$$\tilde{\mathbf{w}}_i(t) = \tilde{\mathbf{R}}^\dagger(t) \tilde{\mathbf{d}}_i(t), \quad i = \overline{1, n}. \quad (27)$$

On the other hand, a block-based adaptive algorithm for designing adaptive filters is one of many efficient adaptive filtering algorithms aimed at increasing convergence speed and reducing the computational complexity just as the block-based least-mean-square (BLMS) algorithm shown in [3, p. 347]. The basic principle of the block-based algorithm for designing an adaptive filter is that the filter coefficients remain unchanged during the processing of each data block and are updated only once per block [3]. Suppose r is the block length. Then the original discrete (or sample) time t is related the k -th block of data as

$$t = kr + i, \quad i = \overline{0, r-1}, \quad k \in \mathbf{Z}. \quad (28)$$

The index k is referred to as the *block index*. Following this principle along with the notation (28), we develop a block-based adaptive multichannel super-exponential algorithm for blind deconvolution of the system (13).

Let k denote the block index. We can rewrite (27) as

$$\tilde{\mathbf{w}}_i(k) = \tilde{\mathbf{R}}^\dagger(k) \tilde{\mathbf{d}}_i(k), \quad i = \overline{1, n}. \quad (29)$$

Then we should obtain recursion formulas for block-updating of matrix $\tilde{\mathbf{R}}(k)$, vector $\tilde{\mathbf{d}}_i(k)$ and pseudoinverse $\tilde{\mathbf{R}}^\dagger(k)$ in (29), respectively.

$$\tilde{\mathbf{R}}(k) = (1 - \alpha_k) \tilde{\mathbf{R}}(k-1) + \alpha_k \tilde{\mathbf{B}}^*(k) \tilde{\mathbf{B}}^T(k), \quad (30)$$

$$\tilde{\mathbf{d}}_i(k) = (1 - \alpha_k) \tilde{\mathbf{d}}_i(k-1) + \alpha_k \sum_{j=0}^{r-1} \tilde{\mathbf{y}}^* \{(k-1)r+j\} \tilde{z}_i \{(k-1)r+j\}, \quad (31)$$

where

$$\tilde{\mathbf{B}}(k) = [\tilde{\mathbf{y}}\{(k-1)r\}, \tilde{\mathbf{y}}\{(k-1)r+1\}, \dots, \tilde{\mathbf{y}}\{(k-1)r+r-1\}] \in \mathcal{C}^{mL \times r}, \quad (32)$$

$$\tilde{z}_i(j) := (|z_i(j)|^2 - 2 < |z_i(j)|^2 > z_i(j) - < z_i^2(j) > z_i^*(j)). \quad (33)$$

Here $< |z_i(j)|^2 >$ and $< z_i^2(j) >$ denote, respectively, the estimates of $E[|z_i(j)|^2]$ and $E[z_i(j)^2]$ at time t , α_k is a positive number close to, but greater than zero, which accounts for some exponential weighting factor or forgetting factor [3]. For example, we may take $\alpha_k = \frac{1}{kr}$.

Because we consider the case when the number of input n is less than the number of output m , i.e., $n < m$, the correlation matrix $\tilde{\mathbf{R}}(k)$ is not of full rank and a singular matrix [9]. Therefore we may apply the matrix pseudo-inversion lemma to the recursive equation (30).

By applying Proposition 1 to (30) for obtaining a recursive formula for block-updating of pseudoinverse $\mathbf{P}(k) = \tilde{\mathbf{R}}^\dagger(k)$, we have the following lemma.

Lemma 1: Let \mathbf{Q} , \mathbf{Q}^\dagger , \mathbf{A} , \mathbf{A}^\dagger , \mathbf{B} , \mathbf{B}_1 and \mathbf{B}_2 in Proposition 1 are respectively defined as

$$\mathbf{Q} = \tilde{\mathbf{R}}(k), \quad (34)$$

$$\mathbf{Q}^\dagger = \mathbf{P}(k) = \tilde{\mathbf{R}}^\dagger(k), \quad (35)$$

$$\mathbf{A} = (1 - \alpha_k) \tilde{\mathbf{R}}(k-1), \quad (36)$$

$$\mathbf{A}^\dagger = \frac{1}{1 - \alpha_k} \tilde{\mathbf{R}}^\dagger(k-1) = \frac{1}{1 - \alpha_k} \mathbf{P}(k-1), \quad (37)$$

$$\mathbf{B} = \mathbf{B}(k) = \sqrt{\alpha_k} \tilde{\mathbf{B}}^*(k), \quad (38)$$

$$\mathbf{B}_1 = \mathbf{B}_1(k) = \tilde{\mathbf{R}}(k-1) \mathbf{P}(k-1) \mathbf{B}(k), \quad (39)$$

$$\mathbf{B}_2 = \mathbf{B}_2(k) = \{\mathbf{I} - \tilde{\mathbf{R}}(k-1) \mathbf{P}(k-1)\} \mathbf{B}(k). \quad (40)$$

Then, substituting these definitions into Proposition 1, the recursion for the pseudoinverse $\mathbf{P}(k) = \tilde{\mathbf{R}}^\dagger(k)$ of the correlation matrix $\tilde{\mathbf{R}}(k)$ from $\mathbf{P}(k-1)$ is explicitly expressed, depending on the values of vectors $\mathbf{B}_1(k)$ and $\mathbf{B}_2(k)$, as follows:

1) If $\mathbf{B}_2(k) = 0$, then

$$\mathbf{P}(k) = \frac{1}{1 - \alpha_k} [\mathbf{P}(k-1) - \mathbf{P}(k-1) \mathbf{B}_1(k) \mathbf{P}_A^{-1}(k) \mathbf{B}_1^H(k) \mathbf{P}(k-1)], \quad (41)$$

where

$$\mathbf{P}_A^{-1}(k) = \{(1 - \alpha_k) \mathbf{I} + \mathbf{B}_1^H(k) \mathbf{P}(k-1) \mathbf{B}_1(k)\}^{-1}. \quad (42)$$

2) If $\mathbf{B}_2(k) \neq 0$ and $\mathbf{B}_1(k) = 0$, then

$$\mathbf{P}(k) = \frac{1}{1 - \alpha_k} \mathbf{P}(k-1) + (\mathbf{B}_2^H(k))^\dagger \mathbf{B}_2^\dagger(k). \quad (43)$$

3) If $\mathbf{B}_2(k) \neq 0$ and $\mathbf{B}_1(k) \neq 0$, then

$$\mathbf{P}(k) = \mathbf{P}_B^\dagger(k) - \mathbf{P}_B^\dagger(k) [\mathbf{B}_1(k), \mathbf{B}_2(k)] \mathbf{P}_D^{-1}(k) [\mathbf{B}_1(k), \mathbf{B}_2(k)]^H \mathbf{P}_B^\dagger(k), \quad (44)$$

where $\mathbf{P}_B^\dagger(k)$ and $\mathbf{P}_D^{-1}(k)$ are respectively defined by

$$\mathbf{P}_B^\dagger(k) := \frac{1}{1 - \alpha_k} [\mathbf{P}(k-1) - \mathbf{P}(k-1) \mathbf{B}_1(k) \mathbf{P}_A^{-1}(k) \mathbf{B}_1^H(k) \mathbf{P}(k-1)] + (\mathbf{B}_2^H(k))^\dagger \mathbf{B}_2^\dagger(k), \quad (45)$$

and

$$\mathbf{P}_D^{-1}(k) := \left[\begin{array}{c|c} -\Delta^{-1}(k) \mathbf{E}_2(k) & \Delta^{-1}(k) \\ \hline \mathbf{I} + \mathbf{E}_1(k) \Delta^{-1}(k) \mathbf{E}_2(k) & -\mathbf{E}_1(k) \Delta^{-1}(k) \end{array} \right] \quad (46)$$

with

$$\Delta(k) := \mathbf{I} - \mathbf{E}_2(k) \mathbf{E}_1(k), \quad (47)$$

where

$$\mathbf{E}_1(k) = \mathbf{B}_1^H(k) \mathbf{P}_B^\dagger(k) \mathbf{B}_1(k), \quad (48)$$

$$\mathbf{E}_2(k) = \mathbf{B}_2^H(k) \mathbf{P}_B^\dagger(k) \mathbf{B}_2(k). \quad (49)$$

These equations are initialized by their values appropriately selected or calculated by the batch algorithm in (19) and (20) at initial block index k_0 and used for $k = k_0 + 1, k_0 + 2, \dots$.

The proof of Lemma 1 is easy and follows from Proposition 1 along with simple calculations.

Based on Lemma 1 along with from (29) through (33), we have following theorem which gives a recursion formula for block-updating of the tap vector $\tilde{\mathbf{w}}_i(k)$ for $i = \overline{1, n}$.

Theorem 1: The recursion for $\tilde{\mathbf{w}}_i(k)$ is

$$\tilde{\mathbf{w}}_i(k) = \mathbf{P}(k) \tilde{\mathbf{R}}(k) \tilde{\mathbf{w}}_i(k-1) + \mathbf{k}(k) \left[\sum_{j=0}^{r-1} \tilde{\mathbf{y}}^* \{(k-1)r+j\} \tilde{z}_i \{(k-1)r+j\} - \tilde{\mathbf{B}}^*(k) \tilde{\mathbf{B}}^T(k) \tilde{\mathbf{w}}_i(k-1) \right], \quad (50)$$

where

$$\mathbf{k}(k) := \alpha_k \mathbf{P}(k), \quad (51)$$

$$\tilde{z}_i(j) := (|z_i(j)|^2 - 2 < |z_i(j)|^2 > z_i(j) - < z_i^2(j) > z_i^*(j)), \quad (52)$$

$$< |z_i(j)|^2 > := (1 - \beta_k) < |z_i(j-1)|^2 > + \beta_k |z_i(j)|^2, \quad (53)$$

$$< z_i^2(j) > := (1 - \beta_k) < z_i^2(j-1) > + \beta_k z_i^2(j). \quad (54)$$

Here β_k is a positive constant greater than α_k , and $\mathbf{P}(k)$ is calculated from (41), (43) or (44) depending on the values of $\mathbf{B}_1(k)$ and $\mathbf{B}_2(k)$.

Remark 2: The recursive algorithm proposed by Shalvi and Weinstein [7] can be shown to correspond to the particular case of Theorem 1 where $n = m = 1$, $r = 1$ and the correlation matrices $\tilde{\mathbf{R}}(k)$'s are nonsingular.

IV. SIMULATION RESULTS

To demonstrate the usefulness of the matrix pseudo-inversion lemma, some computer simulations for obtaining the pseudoinverse $\mathbf{P}(k) = \tilde{\mathbf{R}}^\dagger(k)$ of the correlation matrix $\tilde{\mathbf{R}}(k)$ in (30) by using Lemma 1 were conducted. We note here that we do not use Theorem 1, because we are not interested in finding $\tilde{\mathbf{w}}_i(k)$'s but interested in calculating $\tilde{\mathbf{R}}^\dagger(k)$. The results of calculating $\tilde{\mathbf{w}}_i(k)$'s and recovering original sources $\mathbf{s}(k)$'s will be found in a forthcoming paper.

We considered a MIMO system $\mathbf{H}(z)$ with two inputs ($n = 2$) and five outputs ($m = 5$), and assumed that the system $\mathbf{H}(z)$ is of FIR and the length of channel is three, that is $\mathbf{H}^{(k)}$'s in (14) were set to be

$$\mathbf{H}(z) = \sum_{k=0}^2 \mathbf{H}^{(k)} z^{-k} = \begin{bmatrix} 1.00 + 0.15z^{-1} + 0.10z^{-2} & 0.65 + 0.25z^{-1} + 0.15z^{-2} \\ 0.50 - 0.10z^{-1} + 0.20z^{-2} & 1.00 + 0.25z^{-1} + 0.10z^{-2} \\ 0.60 + 0.10z^{-1} + 0.40z^{-2} & 0.10 + 0.20z^{-1} + 0.10z^{-2} \\ 0.50 + 0.10z^{-1} - 0.20z^{-2} & 0.30 + 0.40z^{-1} + 0.50z^{-2} \\ 0.80 + 0.10z^{-1} + 0.40z^{-2} & 0.10 - 0.20z^{-1} + 0.10z^{-2} \end{bmatrix} \quad (55)$$

Two source signals were 4-PSK and 8-PSK signals, respectively. The length of the equalizer is seven ($L = 7$). For obtaining the pseudoinverse of the correlation matrix, the initial values of $\tilde{\mathbf{R}}$, $\tilde{\mathbf{d}}_i$ and \mathbf{P} were estimated using 30 data samples. The value of α_k was chosen as $\alpha_t = \frac{1}{kr}$ for each k . As a measure of performance, we use the following sum of the Euclidean or Frobenius norms of the four error matrices for each k :

$$M := \frac{\|\mathbf{F}_1(k)\| + \|\mathbf{F}_2(k)\| + \|\mathbf{F}_3(k)\| + \|\mathbf{F}_4(k)\|}{\|\tilde{\mathbf{R}}(k)\|}, \quad (56)$$

where $\|\mathbf{F}_i(k)\|$ denotes the Euclidean or Frobenius norm of matrix $\mathbf{F}_i(k)$, and $\mathbf{F}_1(k)$, $\mathbf{F}_2(k)$, $\mathbf{F}_3(k)$ and $\mathbf{F}_4(k)$ are respectively error

matrices defined by using Moore-Penrose conditions as follows [5].

$$\mathbf{F}_1(k) = \tilde{\mathbf{R}}(k)\mathbf{P}(k)\tilde{\mathbf{R}}(k) - \tilde{\mathbf{R}}(k) \quad (57)$$

$$\mathbf{F}_2(k) = \mathbf{P}(k)\tilde{\mathbf{R}}(k)\mathbf{P}(k) - \mathbf{P}(k) \quad (58)$$

$$\mathbf{F}_3(k) = (\tilde{\mathbf{R}}(k)\mathbf{P}(k))^H - \tilde{\mathbf{R}}(k)\mathbf{P}(k) \quad (59)$$

$$\mathbf{F}_4(k) = (\mathbf{P}(k)\tilde{\mathbf{R}}(k))^H - \mathbf{P}(k)\tilde{\mathbf{R}}(k) \quad (60)$$

In this application, from A2), the input process $\{s(t)\}$ is stationary, and this means that the random process $\{\tilde{\mathbf{y}}^*(t)\}$ is also stationary. Thus it follows from (23) that $\tilde{\mathbf{y}}^*(t)$ belongs almost surely (a.s.) (or with probability 1) to $\text{Im } \tilde{\mathbf{R}}$, that is,

$$\tilde{\mathbf{y}}^*(t) \in \text{Im } \tilde{\mathbf{R}}, \quad a.s. \quad (61)$$

This means from (32), (36) and (38) that

$$R(\mathbf{B}) \subset R(\mathbf{A}). \quad (62)$$

Therefore we can assume in this application that the component matrix $\mathbf{B}_2 = \mathbf{B}_2(k)$ always vanishes, that is, $\mathbf{B}_2(k) = 0$, and we can use the recursion formula (41) for calculating the pseudoinverse $\mathbf{P}(k) = \tilde{\mathbf{R}}^\dagger(k)$ at each iteration (or block index) k . Some results for the case in which the random process $\{\tilde{\mathbf{y}}^*(t)\}$ is not stationary (the time-variant channel), that is, $\mathbf{B}_2(k) \neq 0$, will be presented in the symposium and also found in a forthcoming paper.

We compared the performance of the proposed method (i.e., the method using the matrix pseudo-inversion lemma) with the performance of the method using the built-in function "pinv" in MATLAB Version 7.1.0 for calculating the pseudoinverse $\mathbf{P}(k) = \tilde{\mathbf{R}}^\dagger(k)$ of correlation matrix $\tilde{\mathbf{R}}(k)$. The pseudoinverses are calculated iteratively (or recursively) for each iteration (or recursion) number k for the two methods.

Fig. 1 shows the performance results of the performance measure M for the proposed method with $r = 1$ ((a)) and $r = 2$ ((b)) and for the latter method ((c)) by using 500 data samples with $r = 2$.

We also compared performances of the two methods in computational complexity by using the built-in function "flops" in MATLAB Version 5.2 and in execution time by using a personal computer (PC) with an 3.0 GHz processor and 1GB main memories used in simulation experiments.

Table 1 shows the average of the numbers of floating point operations (flops) and the average of the execution times over 10 independent Monte Carlo runs using 500 data samples of the outputs for each Monte Carlo run for the proposed method (with $r = 1$ and $r = 2$) and the method using "pinv" (with $r = 2$).

It can be seen from Fig. 1 ((a) and (b)) and Table 1 that the performance measure M , the average of the numbers of floating point operations and the average of the execution times of the proposed method are better as the length of the block r increases.

In the meantime, it can be seen from Fig. 1 ((b) and (c)) that the accuracy of the matrix pseudo-inversion lemma is almost equivalent to the built-in function "pinv". However, it can be seen from Table 1 that the average of the numbers of floating point operations and the average of the execution times for the proposed method are better than those for the method using built-in function "pinv" at about 34.2% and 39.1%, respectively.

We consider that one of reasons why the matrix pseudo-inversion lemma is superior to the built-in function "pinv" in the numbers of floating point operations and the execution times is that it is not necessary to calculate the pseudo-inverse \mathbf{A}^\dagger of (4) in Proposition 1, because the results of the previous iteration $\mathbf{P}(k-1) = \tilde{\mathbf{R}}^\dagger(k-1)$ can be used instead of \mathbf{A}^\dagger in (41) of Lemma 1. Therefore, the matrix pseudo-inversion lemma is useful to calculate the pseudoinverse of the matrix for block-based adaptive algorithms of blind deconvolution.

Table 1. Comparison of the averages of the numbers of floating point operations (flops) and the execution times.

The method	flops	times [sec]
The proposed method ($r = 1$)	5.5946×10^8	1.0322
The proposed method ($r = 2$)	2.9405×10^8	0.6171
The method using "pinv" ($r = 2$)	4.4664×10^8	1.0135

V. CONCLUSION

We extended the matrix inversion lemma to the case when the matrix \mathbf{A} in $\mathbf{A} + \mathbf{B}\mathbf{B}^H$ is singular, and presented a matrix pseudo-

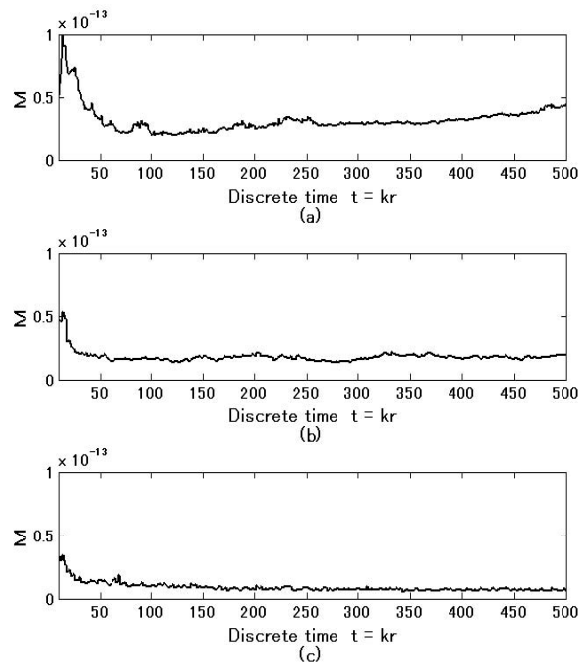


Fig. 1. Performance measure of the M (a) the proposed method ($r = 1$) (b) the proposed method ($r = 2$) (c) the method using "pinv" ($r = 2$).

inversion lemma. In order to show the usefulness of this lemma, we applied it to develop a block-based adaptive super-exponential algorithm for the blind deconvolution of a MIMO system. It has been shown through computer simulations that the matrix pseudo-inversion lemma is useful for block-based adaptive algorithms of blind deconvolution.

ACKNOWLEDGMENT

This work is supported by the Research Projects, Grant-IN-AIDs, No. 1850054[†] and No. 18500146[‡] for Scientific Research of the JSPS.

REFERENCES

- [1] L. Ljung, *System Identification: Theory for the User*. Upper Saddle River, NJ: Prentice-Hall, 1987.
- [2] J. M. Mendel, *Lessons in Estimation Theory for Signal Processing, Communications, and Control*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
- [3] S. Haykin, *Adaptive Filter Theory, 3rd Ed.* Upper Saddle River, NJ: Prentice-Hall, 1996.
- [4] H. Ogawa, "An operator pseudo-inversion lemma," *SIAM Journal on Applied Mathematics*, vol. 48, no. 6, pp. 1527-1531, Dec. 1988.
- [5] S. L. Campbell and C. D. Meyer, Jr., *Generalized Inverses of Linear Transformations*. Dover Publications, Inc., New York, 1979.
- [6] P. Lancaster and M. Tismenetsky, *The Theory of Matrix*. San Diego, CA: Academic, 1985.
- [7] O. Shalvi and E. Weinstein, "Super-exponential methods for blind deconvolution," *IEEE Trans. Information Theory*, vol. 39, no. 2, pp. 504-519, Mar. 1993.
- [8] Y. Inouye and K. Tanebe, "Super-exponential algorithms for multichannel blind deconvolution," *IEEE Trans. Signal Processing*, vol. 48, no. 3, pp. 881-888, Mar. 2000.
- [9] K. Kohno, Y. Inouye, M. Kawamoto and T. Okamoto, "Adaptive super-exponential algorithms for blind deconvolution of MIMO systems," in *Proc. ISCAS 2004*, Vancouver, Canada, May 23-26, 2004, Vol. 5, pp.680-683.
- [10] K. Kohno, Y. Inouye and M. Kawamoto, "An adaptive super-exponential deflation algorithm for blind deconvolution of MIMO systems using the matrix pseudo-inversion lemma," in *Proc. ISCAS 2005*, Kobe, Japan, May 23-26, 2005, pp.5870-5873.