A Block-Based Adaptive Super-Exponential Deflation Algorithm for Blind Deconvolution of MIMO Systems Using the Matrix Pseudo-Inversion Lemma

Kiyotaka Kohno*, Mitsuru Kawamoto† and Yujiro Inouye‡
*Department of Electronic Control Engineering, Yonago National College of Technology,
4448 Hikona-cho, Yonago, Tottori 683-8502, Japan
†Information Technology Research Institute,
National Institute of Advanced Industrial Science and Technology (AIST),
2-41-6 Aomi, Koto-ku, Tokyo 135-0064, Japan
‡Department of Electronic and Control Systems Engineering, Shimane University,
1060 Nishikawatsu, Matsue, Shimane 690-8504, Japan
*kohno@yonago-k.ac.jp, †m.kawamoto@aist.go.jp, ‡inouye@riko.shimane-u.ac.jp

***Abstract*— The matrix inversion lemma gives an explicit formula of the inverse of a positive-definite matrix $\boldsymbol{A}$ added to a block of dyads (represented as $\boldsymbol{B}\boldsymbol{B}^H$). 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 already extended this result to the case when the matrix $\boldsymbol{A}$ is singular, and presented the matrix pseudo-inversion lemma. Such a singular case may occur in a situation where a given problem is overdetermined in the sense that it has more equations than unknowns. In this paper, based on these results, we propose a block-based adaptive multichannel super-exponential deflation algorithm. 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 $\boldsymbol{A}$ added to a block of dyads (represented as $\boldsymbol{B}\boldsymbol{B}^H$) can be represented as

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

where $\boldsymbol{B}$ is an $n \times l$ matrix and the superscript H denotes the complex conjugate transpose (or Hermitian) operation. It is widely 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 $\boldsymbol{A}$ is positive semidefinite [4]. However, his extension is valid under the condition that the range of $\boldsymbol{A}$ includes the range of $\boldsymbol{B}$, but this condition is not satisfied for adaptive signal processing in non-stationary environments.

In the previous paper [5], we extended the matrix inversion lemma in (1) to the case when the matrix $\boldsymbol{A}$ is positive semidefinite without the above condition for the ranges of the relevant matrices, and presented the matrix pseudo-inversion lemma. Based on these results, we applied this lemma to block-based adaptive blind deconvolution of a MIMO system and we proposed the block-based adaptive version of the multichannel super-exponential algorithm for the blind deconvolution. However, we did not present simulation results for recovering source signals, and did not show effectiveness of the block-based adaptive algorithm.

In the present paper, based on the results of [5], after the presentation of the matrix pseudo-inversion lemma, we propose a block-based adaptive version of the multichannel super-exponential deflation algorithm for the blind deconvolution of a MIMO system.

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 7.5 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 $\boldsymbol{C}^{m \times n}$ 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 $\boldsymbol{D} \in \boldsymbol{C}^{n \times l}$ is called a *dyad* (or *dyadic* matrix) if $\boldsymbol{D}$ can be represented as $\boldsymbol{D} = \boldsymbol{b}\boldsymbol{c}^H$ with $\boldsymbol{b} \in \boldsymbol{C}^n$ and $\boldsymbol{c} \in \boldsymbol{C}^l$. Thus a Hermitian dyadic matrix $\boldsymbol{D}$ can be described as $\boldsymbol{D} = \boldsymbol{b}\boldsymbol{b}^H$. The range space (or image) and the null space (or kernel) of $\boldsymbol{E} \in \boldsymbol{C}^{n \times l}$ are denoted by $R(\boldsymbol{A})$ and $N(\boldsymbol{A})$, respectively [7]. Let $i = \overline{1, n}$ stand for $i = 1, 2, \cdots, n$.

II. A Matrix pseudo-Inversion lemma: A general case with a block of dyads

The following lemma gives an explicit formula of the pseudoinverse of a positive semidefinite Hermitian matrix $\boldsymbol{A}$ added to a block of Hermitian dyads (represented as $\boldsymbol{B}\boldsymbol{B}^H$).

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

$$\boldsymbol{B} = \boldsymbol{B}_1 \oplus \boldsymbol{B}_2 \text{ with } R(\boldsymbol{B}_1) \subset R(\boldsymbol{A}) \text{ and } R(\boldsymbol{B}_2) \subset R(\boldsymbol{A})^\perp. \quad (2)$$

Let $\boldsymbol{Q}$ be defined as

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

Then the pseudoinverse $\boldsymbol{Q}^\dagger$ of the matrix $\boldsymbol{Q}$ is explicitly expressed, depending on the values of matrices $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$, as follows:

1) If $\boldsymbol{B}_2 = 0$, then

$$\boldsymbol{Q}^\dagger = \boldsymbol{A}^\dagger - \boldsymbol{A}^\dagger\boldsymbol{B}_1\left(\boldsymbol{I}+\boldsymbol{B}_1^H\boldsymbol{A}^\dagger\boldsymbol{B}_1\right)^{-1}\boldsymbol{B}_1^H\boldsymbol{A}^\dagger. \quad (4)$$

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

$$\boldsymbol{Q}^\dagger = \boldsymbol{A}^\dagger + \left(\boldsymbol{B}_2^H\right)^\dagger\boldsymbol{B}_2^\dagger. \quad (5)$$

3) If $\boldsymbol{B}_2 \neq 0$ and $\boldsymbol{B}_1 \neq 0$, then

$$\boldsymbol{Q}^\dagger = \boldsymbol{Q}_B^\dagger - \boldsymbol{Q}_B^\dagger\left[\boldsymbol{B}_1, \boldsymbol{B}_2\right]\boldsymbol{Q}_D^{-1}\left[\boldsymbol{B}_1, \boldsymbol{B}_2\right]^H\boldsymbol{Q}_B^\dagger, \quad (6)$$

where $\boldsymbol{Q}_B^\dagger$ and $\boldsymbol{Q}_D^{-1}$ are respectively defined by

$$\begin{aligned}\boldsymbol{Q}_B^\dagger &:= \left(\boldsymbol{A}+\boldsymbol{B}_1\boldsymbol{B}_1^H+\boldsymbol{B}_2\boldsymbol{B}_2^H\right)^\dagger \\ &= \boldsymbol{A}^\dagger - \boldsymbol{A}^\dagger\boldsymbol{B}_1\left(\boldsymbol{I}+\boldsymbol{B}_1^H\boldsymbol{A}^\dagger\boldsymbol{B}_1\right)^{-1}\boldsymbol{B}_1^H\boldsymbol{A}^\dagger + \left(\boldsymbol{B}_2^H\right)^\dagger\boldsymbol{B}_2^\dagger, \quad (7)\end{aligned}$$

978-1-4244-5309-2/10/$26.00 ©2010 IEEE

and

$$Q_D^{-1} := \left(P + [B_1, B_2]^H Q_B^{\dagger} [B_1, B_2]\right)^{-1}$$
$$= \left[\begin{array}{c|c} -\Delta^{-1} B_2^H Q_B^{\dagger} B_2 & \Delta^{-1} \\ \hline I + B_1^H Q_B^{\dagger} B_1 \Delta^{-1} B_2^H Q_B^{\dagger} B_2 & -B_1^H Q_B^{\dagger} B_1 \Delta^{-1} \end{array}\right] \in C^{2l \times 2l} \quad (8)$$

with

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

and

$$\Delta := I - B_2^H Q_B^{\dagger} B_2 B_1^H Q_B^{\dagger} B_1. \quad (10)$$

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

Remark 1: A technical important fact in Lemma 1 is that there exists really the inverse of the matrix Q_D defined as

$$Q_D := P + [B_1, B_2]^H Q_B^{\dagger} [B_1, B_2] \quad (11)$$

in (8) even if 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 [7]. This is a key point which is different from the case where B is a column vector [8]. All the proofs of lemma and theorem in this paper are omitted for page limit. They will be appear in a forthcoming paper.

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

Theorem 1: Under the same conditions in Lemma 1, it follows that

$$\begin{aligned} Q^{\dagger} &= (A + BB^H)^{\dagger} \\ &= Q_B^{\dagger} - Q_B^{\dagger} [B_1, B_2] Q_D^{-1} [B_1, B_2]^H Q_B^{\dagger}, \end{aligned} \quad (12)$$

where $Q_B^{\dagger}$ and Q_D^{-1} are defined by (7) and (8), respectively.

III. A Block-Based Adaptive Multichannel Super-Exponential Algorithm

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

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

where $s(t)$ is an n-column vector of input (or source) signals, $y(t)$ is an m-column vector of output signals, and $\{H^{(k)}\}$ is an $m \times n$ matrix sequence of impulse responses.

The transfer function of the channel is defined by

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

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

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

The objective of multichannel blind deconvolution is to construct a deconvolver $W(z)$ that recovers the original source signals only from the measurements of the corresponding outputs. For the time being, it is assumed for theoretical analysis that the noise term n(t) in (13) is absent. However, all the signals and the parameters of the systems are allowed to be complex-valued.

We put the following assumptions on the channel, the source signals and the deconvolver.

A1) The transfer function $H(z)$ 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 $\{s(t)\}$ 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 nonzero variance σ_i^2 and a nonzero fourth-order cumulant γ_i. The variances σ_i^2's and the fourth-order cumulants γ_i's are unknown.

A3) The deconvolver $W(z)$ is an FIR system of sufficient length L so that the truncation effect can be ignored.

Based on assumption A3), let us consider an FIR deconvolver with the transfer function $W(z)$ given by

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

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

$$\tilde{w}_i := \left[w_{i,1}^T, w_{i,2}^T, \cdots, w_{i,m}^T\right]^T \in C^{mL}, \quad (17)$$

$$w_{i,j} = \left[w_{i,j}{}^{(L_1)}, w_{i,j}{}^{(L_1+1)}, \cdots, w_{i,j}{}^{(L_2)}\right]^T \in C^L, \quad (18)$$

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

Inouye and Tanebe [9] proposed the *multichannel super-exponential algorithm* (MSEA) for finding the tap coefficient vectors $\tilde{w}_i$'s of the deconvolver $W(z)$, each iteration of which consists of the following two steps:

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

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

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

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

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

where $y_i(t)$ is the ith element of the output vector $y(t)$ of the channel in (13). Then the correlation matrix $\tilde{R}$ (defined by (41) and (42) in [9]) is represented as

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

and the fourth-order cumulant vector $\tilde{d}_i$ is defined by

$$\tilde{d}_i := \left[d_{i,1}^T, d_{i,2}^T, \cdots, d_{i,m}^T\right]^T \in C^{mL}, \quad (24)$$

whose jth block element $d_{i,j}$ is the L-column vector with rth element $[d_{i,j}]_r$ defined by

$$[d_{i,j}]_r = \text{cum}(z_i(t), z_i(t), z_i^*(t), y_j^*(t-r)) \quad \text{for} \quad r = 0, \pm 1, \pm 2, \cdots, \quad (25)$$

and $\tilde{d}_i$ is represented as

$$\begin{aligned} \tilde{d}_i &= E\left[|z_i(t)|^2 z_i(t) \tilde{y}^*(t)\right] - 2E\left[|z_i(t)|^2\right] E\left[z_i(t)\tilde{y}^*(t)\right] \\ &- E\left[z_i{}^2(t)\right] E\left[z_i{}^*(t)\tilde{y}^*(t)\right] \in C^{mL}, \end{aligned} \quad (26)$$

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

Consider the batch algorithm in (19) and (20). The equation (20) constrains a weighted norm of vector $\tilde{w}_i$ to equal one, and thus we assume this constraint is always satisfied using a normalization or an automatic gain control (AGC) of $\tilde{w}_i$ 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

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

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 l is the block length. Then the original discrete (or sample) time t is related the k-th block of data as

$$t = kl + i, \quad i = \overline{1, l-1}, \quad k \in \boldsymbol{Z}. \tag{28}$$

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

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

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

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

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

where

$$\tilde{\boldsymbol{B}}(k) = [\tilde{\boldsymbol{y}}\{(k-1)l\}, \tilde{\boldsymbol{y}}\{(k-1)l+1\}, \cdots, \tilde{\boldsymbol{y}}\{(k-1)l+l-1\}] \in \boldsymbol{C}^{mL \times l}, \tag{32}$$

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

Here $< |z_i(j)|^2 >$ and $< z_i^2(j) >$ denote respectively the estimates of $E\left[|z_i(j)|^2\right]$ and $E\left[z_i(j)^2\right]$ 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 $\alpha_k = \frac{1}{kl}$.

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

Lemma 2: Let $\boldsymbol{Q}$, $\boldsymbol{Q}^{\dagger}$, $\boldsymbol{A}$, $\boldsymbol{A}^{\dagger}$, $\boldsymbol{B}$, $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ in Lemma 1 are respectively defined as

$$\boldsymbol{Q} = \tilde{\boldsymbol{R}}(k), \tag{34}$$

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

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

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

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

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

$$\boldsymbol{B}_2 = \boldsymbol{B}_2(k) = \left\{\boldsymbol{I} - \tilde{\boldsymbol{R}}(k-1)\boldsymbol{P}(k-1)\right\}\boldsymbol{B}(k). \tag{40}$$

Then, substituting these definitions into Lemma 1, the recursion for the pseudoinverse $\boldsymbol{P}(k) = \tilde{\boldsymbol{R}}^{\dagger}(k)$ of the correlation matrix $\tilde{\boldsymbol{R}}(k)$ from $\boldsymbol{P}(k-1)$ is explicitly expressed, as follows:

$$\boldsymbol{P}(k) = \boldsymbol{P}_B^{\dagger}(k) - \boldsymbol{P}_B^{\dagger}(k)\left[\boldsymbol{B}_1(k), \boldsymbol{B}_2(k)\right] \boldsymbol{P}_D^{-1}(k)\left[\boldsymbol{B}_1(k), \boldsymbol{B}_2(k)\right]^H \boldsymbol{P}_B^{\dagger}(k), \tag{41}$$

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

$$\boldsymbol{P}_B^{\dagger}(k) := \frac{1}{1-\alpha_k}[\boldsymbol{P}(k-1) - \boldsymbol{P}(k-1)\boldsymbol{B}_1(k)\boldsymbol{P}_A^{-1}(k) \boldsymbol{B}_1^H(k)\boldsymbol{P}(k-1)] + \left(\boldsymbol{B}_2^H(k)\right)^{\dagger}\boldsymbol{B}_2^{\dagger}(k), \tag{42}$$

and

$$\boldsymbol{P}_D^{-1}(k) := \left[\begin{array}{c|c} -\boldsymbol{\Delta}^{-1}(k)\boldsymbol{E}_2(k) & \boldsymbol{\Delta}^{-1}(k) \\ \hline \boldsymbol{I} + \boldsymbol{E}_1(k)\boldsymbol{\Delta}^{-1}(k)\boldsymbol{E}_2(k) & -\boldsymbol{E}_1(k)\boldsymbol{\Delta}^{-1}(k) \end{array}\right] \tag{43}$$

with

$$\boldsymbol{\Delta}(k) := \boldsymbol{I} - \boldsymbol{E}_2(k)\boldsymbol{E}_1(k), \tag{44}$$

where

$$\boldsymbol{E}_1(k) = \boldsymbol{B}_1^H(k)\boldsymbol{P}_B^{\dagger}(k)\boldsymbol{B}_1(k), \tag{45}$$

$$\boldsymbol{E}_2(k) = \boldsymbol{B}_2^H(k)\boldsymbol{P}_B^{\dagger}(k)\boldsymbol{B}_2(k). \tag{46}$$

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 $k = k_0+1, k_0+2, \cdots$.

Based on Lemma 2 along with from (29) through (33), we have following theorem which gives a recursion formula for block-updating of the tap vector $\tilde{\boldsymbol{w}}_i(k)$ for $i = \overline{1, n}$.

Theorem 2: The recursion for $\tilde{\boldsymbol{w}}_i(k)$ is

$$\tilde{\boldsymbol{w}}_i(k) = \boldsymbol{P}(k)\tilde{\boldsymbol{R}}(k)\tilde{\boldsymbol{w}}_i(k-1) + \boldsymbol{k}(k) \left[\sum_{j=0}^{r-1} \tilde{\boldsymbol{y}}^*\{(k-1)r+j\}\tilde{z}_i\{(k-1)r+j\} - \tilde{\boldsymbol{B}}^*(k)\tilde{\boldsymbol{B}}^T(k)\tilde{\boldsymbol{w}}_i(k-1)\right], \tag{47}$$

where

$$\boldsymbol{k}(k) := \alpha_k \boldsymbol{P}(k), \tag{48}$$

$$\tilde{z}_i(j) := (|z_i(j)|^2 - 2 < |z_i(j)|^2 >)z_i(j) - < z_i^2(j) > z_i^*(j), \tag{49}$$

$$< |z_i(j)|^2 > := (1-\beta_k) < |z_i(j-1)|^2 > + \beta_k |z_i(j)|^2, \tag{50}$$

$$< z_i^2(j) > := (1-\beta_k) < z_i^2(j-1) > + \beta_k z_i^2(j). \tag{51}$$

Here β_k is a positive constant greater than α_k, and $\boldsymbol{P}(k)$ is calculated from (41).

IV. A Deflation Version of the Adaptive Super-Exponential Algorithm

The *multichannel super-exponential deflation method* (MSEDM) proposed by Inouye and Tanebe [9] uses the second-order correlations to estimate the contributions of an extracted source signal to the channel outputs. Kohno et al. [10] proposed an MSEDM using the higher-order correlations instead of the second-order correlations to reduce the computational complexity in terms of multiplications and to accelerate the performance of equalization. For the details of the MSEDM using the higher-order correlations, see the equations from (15) through (38) in [10]. In the present paper, we proposed a new block-based adaptive multichannel super-exponential deflation algorithm which is an adaptive version of the MSEDM using the higher-order correlations and the matrix pseudo-inversion lemma described in the previous chapter.

In the new algorithm, the following procedures are carried out in each time when channel outputs are observed. Before the following procedures are carried out, it is necessary that $\tilde{\boldsymbol{R}}$, $\tilde{\boldsymbol{d}}_i$, $\tilde{\boldsymbol{w}}_i$ and $\boldsymbol{P}$ are initialized. At first, set $k = k_0$, and set $l_k = 1$ where l_k denotes the number of channels (or the sources) equalized. Then, $\tilde{\boldsymbol{R}}(k)$ is calculated by (30), $\tilde{\boldsymbol{d}}_{l_k}(k)$ is calculated by using (31), (49), (50) and (51), $\boldsymbol{P}(k)$ is calculated by using from (38) to (46), and $\tilde{\boldsymbol{w}}_{l_k}(k)$ is calculated by the two steps (47) and (20). By these procedures, the first equalized output $z_{l_k}(t)$ $(t = kl+i, i = \overline{1, l-1})$ is obtained.

Next, the MSEDM using the higher-order correlations is carried out. We calculate the contribution signals by using the equalized output $z_{l_k}(t)$, and remove the contribution signals from the channel outputs in order to define the outputs of a multichannel with $n-1$ inputs and m outputs. The number of inputs becomes deflated by one. The procedures mentioned above are continued until $l_k = n$, where we obtain the last equalized output $z_n(t)$ for $k = k_0$. If $k < k_f$ (where k_f is a final block), then set $k = k_0 + 1$ and iterate the same procedures as the previous block k. If $k = k_f$, then stop here. The n equalized outputs $z_1(t), \cdots, z_n(t)$ are obtained for $k = k_0$, $k_0+1, \cdots, k_f$. Therefore, the proposed algorithm is summarized as shown in Table 1.

Table 1. The proposed algorithm.

Step	Contents
1	Set $k = k_0$ (where k_0 is an initial time).
2	Set $l_k = 1$ (where l_k denotes the number of the channels equalized).
3	Calculate $\tilde{\boldsymbol{R}}(k)$ using (30).
4	Calculate $\tilde{\boldsymbol{d}}_{l_k}(k)$ using (31), (49), (50) and (51).
5	Calculate $\boldsymbol{P}(k)$ using from (38) through (46).
6	Calculate $\tilde{\boldsymbol{w}}_{l_k}(k)$ using (47) and (20).
7	Carry out the deflationary process using the MSEDM with the higher-order correlations [10].
8	If the subscript l_k is less than n, then replace l by $l+1$, and the procedures (from Step 3 through Step 7) are continued until $l_k = n$.
9	If $k < k_f$ (where k_f is a final block), then replace k by $k+1$ and iterate the procedures from Step 2 through Step 8. If $k = k_f$, then stop here.

V. Simulation Results

To demonstrate the effectiveness of proposed algorithm, some computer simulations were conducted. We considered a MIMO system $\boldsymbol{H}(z)$ with two inputs ($n = 2$) and three outputs ($m = 3$), and assumed that the system $\boldsymbol{H}(z)$ is of FIR and the length of channel is three ($K = 3$), that is $\boldsymbol{H}^{(k)}$'s in (14) were set to be

$$\boldsymbol{H}(z) = \sum_{k=0}^{2} \boldsymbol{H}^{(k)} z^{-k} = \begin{bmatrix} 1.00 + 0.15z^{-1} + 0.10z^{-2} & 0.65 + 0.25z^{-1} + 0.15z^{-2} \\ 0.50 - 0.10z^{-1} + 0.20z^{-2} & 1.00 + 0.25z^{-1} + 0.10z^{-2} \\ 0.60 + 0.10z^{-1} + 0.40z^{-2} & 0.10 + 0.20z^{-1} + 0.10z^{-2} \end{bmatrix}. \quad (52)$$

The length of the deconvolver was chosen to be six ($L = 6$). We set the values of the tap coefficients to be zero expect for $w_{13}^{(3)}$ = $w_{22}^{(3)}$ = 1. Two source signals were 4-PSK and 8-PSK signals, respectively. For recovering the source signals, initial values of $\tilde{\boldsymbol{R}}$, $\tilde{\boldsymbol{d}}_i$ and $\boldsymbol{P}$ were estimated using 2000 data samples. The value of α_k was chosen as $\alpha_t = \frac{1}{kl}$ for each k. As a measure of performance, we use the *multichannel intersymbol interference* ($\mathrm{M_{ISI}}$) [9].

Fig. 1 shows the performance results of $\mathrm{M_{ISI}}$ for the proposed method (i.e., the method using the matrix pseudo-inversion lemma) with $l = 2$ for the time-variant channel obtained by using 30,000 data samples. In Fig. 1, the second matrix $\boldsymbol{H}^{(1)}$ of the impulse response of the channel was varied by adding 0.05 to all its elements at discrete time t =10,000.

It can be seen from Fig. 1 that the algorithm using the matrix pseudo-inversion lemma is effective for the time-variant channel.

We also compared the performance of the proposed method with the performance of the method using the built-in function "pinv" in MATLAB Version 7.5. The tap vector $\tilde{\boldsymbol{w}}_i(t)$ was calculated in each k by using (47) for the proposed method, while the tap vector $\tilde{\boldsymbol{w}}_i(t)$ was calculated in each k by using (29) for the method using "pinv".

Table 2 shows the averages of the execution times per a Monte Carlo run over 10 independent Monte Carlo runs on a personal computer (Windows machine) with a 2.13 GHz processor and 1GB main memories used in simulation experiments. In each Monte Carlo run, we carried out the simulations using 5,000 data samples of the channel outputs with l = 1, 2, 3, 4 and 5.

It can be seen from Table 2 that the averages of the execution times for the proposed method are better as the length of the block l increases, and are better than those for the method using built-in function "pinv" at about 12.7% ($l = 2$) and 5.54% ($l = 5$).

Therefore, the matrix pseudo-inversion lemma is useful to calculate the pseudoinverse of the matrix for block-based adaptive algorithms of blind deconvolution.

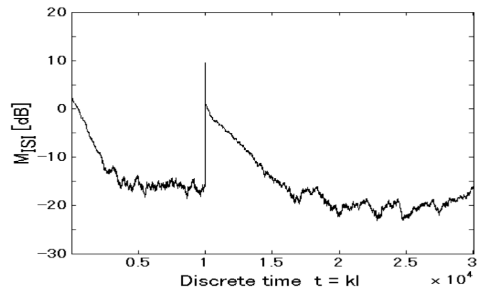

Fig. 1. Performance results of the proposed method ($l = 2$).

Table 2. Comparison of the execution times [sec].

The block length	the proposed method	the method using "pinv"
$l = 1$ (non-block-based)	5.6546	7.5147
$l = 2$	5.1453	5.8959
$l = 3$	4.9208	5.4662
$l = 4$	4.8056	5.1960
$l = 5$	4.7479	5.0264

VI. Conclusions

We presented the matrix pseudo-inversion lemma, applied it to develop an block-based adaptive super-exponential algorithm, and proposed a block-based adaptive version of the multichannel super-exponential deflation 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. The important remaining issue is the stability analysis of the algorithm.

Acknowledgment

This work was supported by the Japan Society of the Promotion of Science (JSPS) Research Projects under Grant-in-Aid 21500088.

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] K. Kohno, Y. Inouye and M. Kawamoto, "A matrix pseudo-inversion lemma and its application to block-based adaptive blind deconvolution for MIMO systems," in *Proc. ISCAS 2007*, New Orleans, U.S.A., May 27-30, 2007, pp.3490-3493.

[6] P. Lancaster and M. Tismenetsky, *The Theory of Matrix*. San Diego, CA: Academic, 1985.

[7] S. L. Campbell and C. D. Meyer, Jr., *Generalized Inverses of Linear Transformations*. Dover Publications, Inc., New York, 1979.

[8] K. Kohno, Y. Inouye and M. Kawamoto, "A matrix pseudo-inversion lemma for positive semidefinite hermitian matrices and its application to adaptive blind deconvolution of MIMO systems," *IEEE Trans. Circuits and Systems-I*, vol. 55, no. 1, pp. 424-435, Feb. 2008.

[9] 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.

[10] K. Kohno, Y. Inouye and M. Kawamoto, "A super-exponential deflation method incorporated with higher-order correlations for blind deconvolution of MIMO linear systems," *Signal Processing*, vol. 86, Issue 11, pp. 3505-3512, Nov. 2006.