Scientific
Research

A Modified Eigenvector Method for Blind Deconvolution of MIMO Systems Using the Matrix Pseudo-Inversion Lemma*

Mitsuru Kawamoto[1], Kiyotaka Kohno[2], Yujiro Inouye[3], Koichi Kurumatani[1]

[1]*Information Technology Research Institute*, *National Institute of Advanced Science and Technology*, Tsukuba, *Japan*
[2]*Department of Electronic Control Engineering*, *Yonago National College of Technology*, *Yonago-city*, *Japan*
[3]*Department of Electronic and Control Systems Engineering*, *Shimane University*, *Matsue*, *Japan*
E-mail: *{m.kawamoto, k.kurumatani}@aist.go.jp, kohno@yonago-k.ac.jp, inouye@riko.shimane-u.ac.jp*

Abstract

Recently we have developed an eigenvector method (EVM) which can achieve the blind deconvolution (BD) for MIMO systems. One of attractive features of the proposed algorithm is that the BD can be achieved by calculating the eigenvectors of a matrix relevant to it. However, the performance accuracy of the EVM depends highly on computational results of the eigenvectors. In this paper, by modifying the EVM, we propose an algorithm which can achieve the BD without calculating the eigenvectors. Then the pseudo-inverse which is needed to carry out the BD is calculated by our proposed matrix pseudo-inversion lemma. Moreover, using a combination of the conventional EVM and the modified EVM, we will show its performances comparing with each EVM. Simulation results will be presented for showing the effectiveness of the proposed methods.

1. Introduction

In this paper, we deal with a blind deconvolution (BD) problem for a multiple-input and multiple-output (MIMO) infinite-impulse response (IIR) channels. A large number of methods for solving the BD problem have been proposed until now (see [1], and reference therein). In order to solve the BD problem, this paper focuses on the eigenvector method (EVM).

The first proposal of the EVM was done by Jelonnek *et al.* [2]. They have proposed the EVM for solving blind equalization (BE) problems of single-input single-output (SISO) channels and single-input multiple-output (SIMO) channels. The most attractive feature of the EVM is that its algorithm can be derived from a closed-form solution using reference signals. Then, a generalized eigenvector problem can be formulated and the eigenvector calculation is carried out in order to solve the BE problem. Owing to the property, differently from the algorithms derived from steepest descent methods, the EVM does not need many iterations to achieve the BE, but works so as to solve the BE problem

with one iteration.

Recently, we extended the EVM to the case of MIMO-IIR channels [3,4]. Then we proved that the proposed EVM can work so as to recover all source signals from their mixtures with one iteration. However, in the EVM, its performance accuracy depends highly on computational results of the eigenvectors.

In this paper, we modify the EVM and then an algorithm for solving the BD is proposed, in which the proposed algorithm can be carried out without calculating the eigenvectors. Namely, the proposed algorithm can achieve the BD with as less computational complexity as possible, compared with the conventional EVMs. Moreover, a combination of the conventional EVM and the modified EVM is proposed. The combined EVM has such properties that the BD can be achieved with as less computational complexity as possible and with good accuracy compared with each EVM.

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^n denote the set of all n-column vectors with complex components. Let $C^{m \times n}$ denote the set of all $m \times n$ matrices with complex components. The super-

---

scripts T, $*$, and H denote, respectively, the transpose, the complex conjugate, and the complex conjugate transpose (Hermitian) of a matrix. The symbol † denotes a pseudo-inverse of a matrix. The symbols block-diag $\{\cdots\}$ and diag $\{\cdots\}$ denote respectively a block diagonal and a diagonal matrices with the block diagonal and the diagonal elements $\{\cdots\}$. The symbol cum$\{x_1, x_2, x_3, x_4\}$ denotes the fourth-order cumulant of x_i's. Let $i = \underline{1, n}$ stand for $I = 1, 2, \cdots, n.$

2. Problem Formulation and Assumptions

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

$$y(t) = \sum_{k=-\infty}^{\infty} H^{(k)}s(t-k) + n(t), \ t \in Z, \qquad (1)$$

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

The transfer function of the system is defined by $H(z) = \sum_{k=-\infty}^{\infty} H^{(k)}z^k$, $z \in C$.

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

$$v(t) = \sum_{k=-\infty}^{\infty} W^{(k)}y(t-k)$$

$$= \sum_{k=-\infty}^{\infty} G^{(k)}s(t-k) + \sum_{k=-\infty}^{\infty} W\,n(t-k), \qquad (2)$$

where $\{G^{(k)}\}$ is the impulse response matrix sequence of $G(z) := W(z)H(z)$, which is defined by $G(z) = \sum_{k=-\infty}^{\infty} G^{(k)}z^k$, $z \in C$. The cascade connection of the unknown system and the deconvolver is illustrated in **Figure 1.**

Here, we put the following assumptions on the system, the source signals, the deconvolver, and the noises.

A1) The transfer function $H(z)$ is stable and has full column rank on the unit circle $|z| = 1$, where the assumption **A1)** implies that the unknown system has less inputs than outputs, *i.e.*, $n \le m$, and there exists a left stable inverse of the unknown system. Please do not revise any of the current designations.
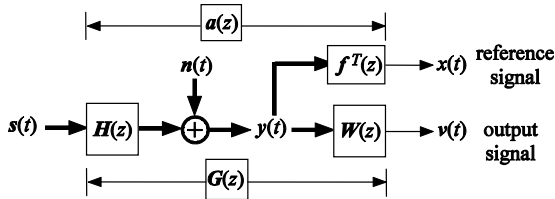

Figure 1. The composite system of the unknown system $H(z)$ and the deconvolver $W(z)$, and the reference system $f(z)$ with m inputs and single output $x(t)$. It is the case of single reference.

A2) The input sequence $\{s(t)\}$ is a complex, zero-mean and non-Gaussian random vector process with element processes $\{s_i(t)\}$, $i = 1, n$ being mutually independent. Each element process $\{s_i(t)\}$ is an i.i.d. process with a variance $\sigma_{s_i}^2 \neq 0$ and a nonzero fourth-order cumulant $\gamma_i \neq 0$ defined as

$$\gamma_i = \text{cum}\,s_i\left\{(t), s_i(t), s_i^*(t), s_i^*(t)\right\} \neq 0. \qquad (3)$$

A3) The deconvolver $W(z)$ s an FIR system, that is, $W(z) = \sum_{k=L_1}^{L_2} W^{(k)}z^k$, where the length $L := L_2 - L_1 + 1$ is taken to be sufficiently large so that the truncation effect can be ignored.

A4) The noise sequence $\{n(t)\}$ is a zero-mean, Gaussian vector stationary process whose component processes $\{n_j(t)\}$, $j = \underline{1, m}$ have nonzero variances $\sigma_{n_i}^2 \neq 0$, $j = \underline{1, m}$.

A5) The two vector sequences $\{n(t)\}$ and $\{s(t)\}$ are mutually statistically independent.

Under **A3)**, the impulse response $\{G^{(k)}\}$ of the cascade system is given by

$$G^{(k)} := \sum_{\tau=L_1}^{L_2} W^{(\tau)}H^{(k-\tau)}, \ k \in Z \qquad (4)$$

In a vector form, (4) can be written as

$$\tilde{g}_i = \tilde{H}\tilde{w}_i, \ i = 1, n \qquad (5)$$

where $\tilde{g}_i$ is the column vector consisting of the i-th output impulse response of the cascade system defined by $\tilde{g}_i := \left[g_{i1}^T, g_{i2}^T, \cdots, g_{in}^T\right]^T$, g_{ij} is expressed as

$$g_{ij} := \left[\cdots, g_{ij}(-1), g_{ij}(0), g_{ij}(1), \cdots\right]^T, j = 1, n \qquad (6)$$

where $g_{ij}(k)$ is the (i, j)-th element of matrix $G^{(k)}$, and $\tilde{w}_i$ is the mL-column vector consisting of the tap coefficients (corresponding to the i-th output) of the deconvolver defined by $\tilde{w}_i := [w_{i1}^T, w_{i2}^T, \cdots, w_{in}^T]^T \in C^{mL}$, w_{ij} is defined by

$$w_{ij} := \left[w_{ij}(L_1), w_{ij}(L_1+1), \cdots, w_{ij}(L_2)\right]^T \in C^L, j = 1, m, \quad (7)$$

where $w_{ij}(k)$ is the (i,j)-th element of matrix $W^{(k)}$, and $\tilde{H}$ is the $n \times m$ block matrix whose (i,j)-th block element H_{ij} is the matrix (of L columns and possibly infinite number of rows) with the (l,r)-th element $[H_{ij}]_{lr}$ defined by $[H_{ij}]_{lr} := h_{ji}(l - r)$, $l = 0, \pm 1, \pm 2, \cdots$, $r = \underline{L_1, L_2}$, where $h_{ij}(k)$ is the (i,j)-th element of the matrix $H^{(k)}$.

In the MIMO deconvolution problem, we want to adjust $\tilde{w}_i$'s ($i = \underline{1, n}$) so that

$$[\tilde{g}_1, \cdots, \tilde{g}_n] = \tilde{H}[\tilde{w}_1, \cdots, \tilde{w}_n] = \left[\tilde{\delta}_1, \cdots, \tilde{\delta}_n\right]P, \qquad (8)$$

where P is an $n \times n$ permutation matrix, and $\tilde{\delta}_i$ is the

n-block column vector defined by $\tilde{\delta}_i = \left[\delta_{i1}^T, \delta_{i2}^T, \cdots, \delta_{in}^T\right]^T$, $i = \underline{1,n}$, $\delta_{ij} := \hat{\delta}_i$ for $i = j$, otherwise. $(\cdots, 0, 0, 0, \cdots)^T$ Here, $\hat{\delta}_i$ is the column vector (of infinite elements) whose r-th element $\hat{\delta}_i(\tau)$ is given by $\hat{\delta}_i(\tau) = d_i \delta(r - k_i)$, where $\delta(t)$ is the Kronecker delta function, d_i is a complex number standing for a scale change and a phase shift, and k_i is an integer standing for a time shift.

3. The Conventional Eigenvector Algorithms

Jelonnek *et al.* [2] have shown in the single-input case that from the following problem, that is, Maximize $D_{v_i x} = \text{cum}\left\{v_i(t), v_i^*(t), x(t), x^*(t)\right\}$

under $\sigma_{v_i}^2 = \sigma_{s\rho_i}^2$,   (9)

a closed-form solution expressed as a generalized eigenvector problem can be led by the Lagrangian method, where $\sigma_{v_i}^2$ and $\sigma_{s\rho_i}^2$ denote the variances of the output $v_i(t)$ and a source signal $s_{\rho_i}(t)$, respectively, ρ_i is one of integers $\{1, 2 \cdots, n\}$ such that the set $\{\rho_1, \rho_2, \cdots, \rho_n\}$ is a permutation of the set $\{1, 2, \cdots, n\}$, $v_i(t)$ is the i-th element of $v(t)$ in (2), and the reference signal $x(t)$ is given by $f^T(z)y(t)$ using an appropriate filter $f(z)$ (see **Figure 1**). The filter $f(z)$ is called a *reference system*. Let $a(z) := H^T(z)f(z) = \left[a_1(z), a_2(z), \cdots, a_n(z)\right]^T$, then $x(t) = f^T(z)H(z)s(t) = a^T(z)s(t)$. The element $a_i(z)$ of the filter $a(z)$ is defined as $a_i(z) = \sum_{k=-\infty}^{\infty} a_i(k)z^k$ and the reference system $f(z)$ is an m-column vector whose elements are $f_j(z) = \sum_{k=L_1}^{L_2} f_j(k)z^k$, $j = \underline{1,m}$, where differently from the $w_{ij}(k)$, the parameter $f_j(k)$ is any fixed value.

In our case, $D_{v_i x}$ and $\sigma_{v_i}^2$ can be expressed in terms of the vector $\tilde{w}_i$ as, respectively, $D_{v_i x} = \tilde{w}_i^H \tilde{B} \tilde{w}_i$ and $\sigma_{v_i}^2 = \tilde{w}_i^H \tilde{R} \tilde{w}_i$ where $\tilde{B}$ is the $m \times m$ block matrix whose (i,j)-th block element B_{ij} is the matrix with the (l,r)-th element calculated by cum $\left\{y_j(tL_1 r + 1), y_i^*(tL_1 l + 1), x^*(t), x(t)\right\}$ $(l, r = \underline{1,L})$ and $\tilde{R} = E\left[\tilde{y}^*(t)\tilde{y}^T(t)\right]$ is the covariance matrix of m-block column vector $\tilde{y}(t)$ defined by

$$\tilde{y}(t) := \left[y_1^T(t), y_2^T(t), \ldots, y_m^T(t)\right]^T \in \mathbb{C}^{mL} \quad (10)$$

where

$y_j(t) := \left[y_j(tL_1), y_j(tL_1 1), \cdots, y_j(tL_2)\right]^T \in \mathbb{C}^L$, $j = \underline{1,m}$. It follows from (10) that $\tilde{y}(t)$ is expressed as $\tilde{y}(t) = D_c(z)y(t)$, where $D_c(z)$ is an $mL \times m$ converter (consisting of m identical delay chains each of which has L delay elements when $L_1 = 1$) defined by $D_c(z) :=$ block-diag $\{d_c(z), \cdots, d_c(z)\}$ with m diagonal block elements all being the same L-column vector $d_c(z)$ defined by $d_c(z) = \left[z^{L_1}, \cdots, z^{L_2}\right]^T$. Therefore, by the similar way to as in [2], the maximization of $|D_{v_i x}|$ under $\sigma_{v_i}^2 = \sigma_{s\rho_i}^2$ leads to the following generalized eigenvector problem;

$$\tilde{B}\tilde{w}_i = \lambda_i \tilde{R}\tilde{w}_i. \quad (11)$$

Moreover, Jelonnek et al. have shown in [2] that the eigenvector corresponding to the maximum magnitude eigenvalue of $\tilde{R}^\dagger \tilde{B}$ becomes the solution of the blind equalization problem, which is referred to as an *eigenvector algorithm* (EVA). It has been also shown in [3] that the BD for MIMO-IIR systems can be achieved with the eigenvectors of $\tilde{R}^\dagger \tilde{B}$, using only one reference signal. Note that since Jelonnek *et al.* have dealt with SISO-IIR systems or SIMO-IIR systems, the constructions of $\tilde{B}$, $\tilde{w}_i$, and $\tilde{R}$ in (11) are different from those proposed in [2].

Castella *et al.* [6] have shown that from (9), a BD can be iteratively achieved by using $x_i(t) = \tilde{w}_i \tilde{y}(t)$ $(i = \underline{1,n})$ as reference signals (see **Figure 2**), where the number of reference signals corresponds to the number of source signals and $\tilde{w}_i$ is a vector obtained by $\tilde{R}^\dagger \tilde{B}_i$ divided by λ_i in the previous iteration, where $\tilde{B}_i$ represents $\tilde{B}$ in (11) calculated by $x_i(t) = \tilde{w}_i \tilde{y}(t)$. Namely, they considered the following equation;

$$\tilde{R}^\dagger \tilde{B}_i \tilde{w}_i = \lambda_i \tilde{w}_i. \quad (12)$$

Then a deflation method was used to recover all source signals. However, the EVM proposed by Castella *et al.* requires the calculation of the eigenvectors of the matrix $\tilde{R}^\dagger \tilde{B}_i$ to achieve the BD.

4. The Proposed Algorithm

Here, the equation (12) can be interpreted as follows. Suppose that the value $\tilde{w}_i$ in the left-hand side of (12) is a vector obtained by $\tilde{R}^\dagger \tilde{B}_i$ divided by λ_i in the previous iteration. Also, let $\tilde{d}_i$ denote $\tilde{B}_i \tilde{w}_i$. Then (12) can be expressed as
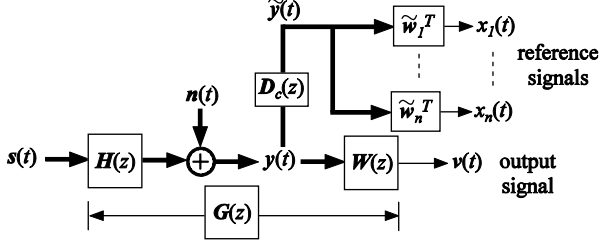
         CS

Figure 2. The composite system of the unknown system *H(z)* and the deconvolver *W(z)*, and the reference system with *m* inputs and *n* outputs, where *Dc(z)* is an *mL×m* converter. It is the case of multiple reference system.

$$\widetilde{w}_i = \frac{1}{\lambda_i} \widetilde{R}^\dagger \widetilde{d}_i, \quad i = \underline{1,n}, \tag{13}$$

where on the details of $\widetilde{d}_i = \widetilde{B}_i \widetilde{w}_i$, see (30) in Appendix. Differently from the EVM in [6], (13) means that $\widetilde{w}_i$ is modified iteratively by the value of the right-hand side of (13) without calculating the eigenvectors of $\widetilde{R}^\dagger \widetilde{B}_i$ where $\widetilde{w}_i$ in both $x_i(t)$ and $\widetilde{d}_i$ is the value of the left-hand side of (13) in the previous iteration. Moreover, the EVMs in [2,5] must select the appropriate parameter for the reference system *f(z)*, but our proposed algorithm does not need such a troublesome process. The scalar λ_i is fixed to be 1, but $\widetilde{w}_i$ obtained by (13) should be normalized at each iteration, that is

$$\widetilde{w}_i := \frac{\widetilde{w}_i}{\sqrt{\widetilde{w}_i^H \widetilde{R} \widetilde{w}_i}}, \quad i = 1,n. \tag{14}$$

It can be seen that the iterative algorithm (13) is nothing but an iterative procedure of the super-exponential method (SEM) [7,8,9] (see Appendix), where the first proposal of the SEM was done by Shalvi and Weinstein [9]. Therefore, our proposed algorithm for achieving the BD is that the vector $\widetilde{w}_i$ is modified by using the value $\widetilde{R}^\dagger \widetilde{d}_i$ in (13), and then the modified vector, that is, $\widetilde{w}_i$ in the left-hand side of (13) is normalized by (14).

Here, the calculation of $\widetilde{R}^\dagger$ is implemented by using the following algorithm based on the matrix pseudo-inversion lemma proposed in [10]. The reason is that in the case that the pseudo-inverse is calculated using data block, the convergence speed is increased and the computational complexity is reduced, compared with the conventional pseudo-inverse algorithms, for example, the built-in function "pinv" in MATLAB [11]. Therefore, in order to provide a recursive formula based on block data for time-updating of pseudo-inverse, the block index "k" is defined, and then $\widetilde{R}$ and $\widetilde{R}^\dagger$ are described as $\widetilde{R}(k)$ and $P(k)$, respectively, where the k-th block of data is defined as

$$t = kl + i, \quad i = \underline{1,l-1}, \quad k \in Z \tag{15}$$

the parameters *l* and *t* denote the block length and the

original discrete (or sample) time, respectively. The matrix $\widetilde{R}(k)$ is obtained by

$$\widetilde{R}(k) = (1-\alpha_k)\widetilde{R}(k-1) + \alpha_k Y^*(k)Y^T(k), \tag{16}$$

where

$$Y(k) = \left[\tilde{y}\{(k-1)l\}, \tilde{y}\{(k-1)l+1\}, \cdots, \tilde{y}\{(k-1)l+l-1\} \right] \in C^{mL \times l} \tag{17}$$

and α_k is a positive number close to, but greater than zero, which accounts for some exponential weighting factor or forgetting factor [13]. Moreover, the following parameters are defined;

$$Y(k) = \sqrt{\alpha_k} \tilde{Y}^*(k), \tag{18}$$

$$Y_1(k) = \tilde{R}(k-1)P(k-1)Y(k), \tag{19}$$

$$Y_2(k) = \{I - \tilde{R}(k-1)P(k-1)\}Y(k). \tag{20}$$

Then the pseudo-inverse $P(k)$ can be explicitly expressed, as follows:

$$P(k) = P_B^\dagger(k) -$$
$$P_B^\dagger(k)\left[Y_1(k), Y_2(k)\right]P_D^{-1}(k)\left[Y_1(k), Y_2(k)\right]^H P_B^\dagger(k) \tag{21}$$

where $P_B^\dagger(k)$ and $P_D^{-1}(k)$ are respectively defined by

$$P_B^\dagger(k) := \frac{\left[P(k-1) - P(k-1)Y_1 k P_A^{-1} Y_1^H(k)P(k-1)\right]}{1-\alpha_k} + \left(Y_2^H(k)\right)^\dagger Y_2^\dagger(k), \tag{22}$$

and

$$P_D^{-1}(k) := \begin{bmatrix} -\Delta^{-1}(k)P_2(k) & \Delta^{-1}(k) \\ I + E_1(k)\Delta^{-1}(k)E_2(k) & -E_1(k)\Delta^{-1}(k) \end{bmatrix} \tag{23}$$

with

$$\Delta(k) := I - E_2(k)E_1(k), \tag{24}$$

where

$$E_1(k) = B_1^H(k)P_B^\dagger(k)B_1(k), \tag{25}$$
$$E_2(k) = B_2^H(k)P_B^\dagger(k)B_2(k),$$

We treat $P(k)$ as $\widetilde{R}^\dagger$, and $\widetilde{w}_i$ is iteratively modified using (13) and (14), where λ_i in (13) is assumed to be fixed to 1 and $\widetilde{d}_i := \widetilde{B}_i \widetilde{w}_i$ in (13) is estimated by using $Y(k)$.

Thus, the proposed iterative algorithm for solving the BD problem is summarized, as follows:

1) Choose appropriate initial values of $\widetilde{w}_i(0)$, $P(0)$, $\widetilde{R}(0)$, $\widetilde{d}_i(0)$, $i=\underline{1,n}$ and set k = 1.

2) Estimate $\widetilde{R}(k-1)$, $\tilde{d}_i(k-1)$, by their moving averages, and $P(k-1)$ by (21).

     CS

3) Calculate the $\widetilde{\boldsymbol{w}}_i(\mathrm{k})$, from $\boldsymbol{P}(\mathrm{k}\text{-}1)\widetilde{\boldsymbol{w}}_i(\mathrm{k})$, and then $\widetilde{\boldsymbol{w}}_i(\mathrm{k})$ is normalized by $\sqrt{\widetilde{\mathrm{w}}_i^H(\mathrm{k})\widetilde{\mathrm{R}}(\mathrm{k}-1)\widetilde{\mathrm{w}}_i(\mathrm{k})}$.

4) Put k = k + 1 and stock the $\widetilde{w}_i$ obtained in (13).

If k = k' (where k' denotes an appropriate iteration number), stop the iterations, otherwise go to **2)**.

5. Simulation Results

To demonstrate the proposed algorithm, 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 FIR and the length of channel is three, that is $\boldsymbol{H}^{(k)}$'s in (1) were set to be

$$\boldsymbol{H}(z) = \sum_{k=0}^{2} \boldsymbol{H}^{(k)} z^k$$

$$= \begin{bmatrix} 1+0.15z+0.1z^2 & 0.65+0.25z+0.15z^2 \\ 0.5-0.1z+0.2z^2 & 1+0.25z+0.1z^2 \\ 0.6+0.1z+0.4z^2 & 0.1+0.2z+0.1z^2 \end{bmatrix}$$

The source signals $s_1(t)$ and $s_2(t)$ were a sub-Gaussian signal which takes one of two values, -1 and 1 with equal probability 1/2. The parameters L_1 and L_2 in $\boldsymbol{W}(z)$ were set to be 0 and 9, respectively. As a measure of performances, we used the *multichannel intersymbol interference* (M_{ISI}) [8], which was the average of 50 Monte Carlo runs. In each Monte Carlo run, using 300 data samples, $\widetilde{w}_i$ is modified by (13) and (14), and the total number of modification times is 10. About the block length l, the following two cases were considered: $l = 1$ and $l = 2$. For obtaining the pseudo-inverse of the correlation matrix, the initial values of $\widetilde{\boldsymbol{R}}$, $\widetilde{\boldsymbol{d}}_i$, and $\boldsymbol{P}$ were estimated using 30 data samples. The value of α_{k} was chosen as $\alpha_{\mathrm{k}} = \dfrac{1}{\mathrm{k}\,l}$ for each k.

Figure 3 shows the results obtained by the conventional EVM (ConEVM), the modified EVM (ModEVM), and their combined EVM (ComEVM) in the case of $l = 1$. As a ConEVM, we selected the EVM proposed by Castella *et al.*. Then, the pseudo-inverse of $\widetilde{\boldsymbol{R}}$ in (12) was calculated by the built-in function "pinv" in MATLAB and our proposed matrix pseudo-inversion lemma, denoted by "mpinvl". In the ComEVM, the ConEVM was carried out at the first modification and from the second modification the ModEVM was carried out, where the pseudo-inverse $\widetilde{\boldsymbol{R}}$ in (12) was calculated by "mpinvl". From the figure, the ConEVM with mpinvl provides a better performance compared with the other EVMs, except for the ComEVM. However, the average of the execution time of the ConEVM with mpinvl is longer than the one of the ModEVM with mpinvl (see **Table 1**).
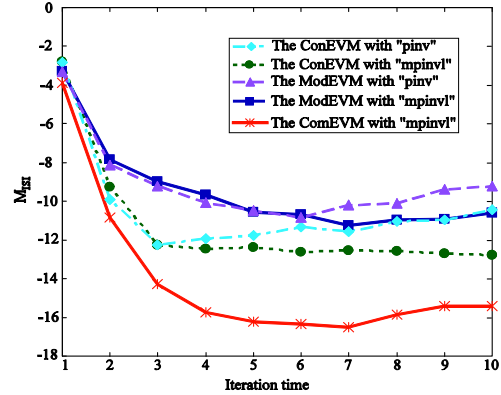

Figure 3. The performances of the proposed algorithm and the conventional methods ($l = 1$).

On the other hand, the ComEVM with mpinvl is carried out with a little bit longer execution time than the ModEVM with mpinvl but the performance of the ComEVM with mpinvl is better than the other EVMs. From these results, we recommend to use the ComEVM with mpinvl to achieve the BD in the case of $l = 1$.

Figure 4 shows the results obtained by the EVMs in the case of $l = 2$. From **Figure 4**, one can see that the ModEVMs with mpinvl provides better performances than the other EVMs. Therefore we recommend to use the ModEVM with mpinvl to achieve the BD in the case of $l = 2$.

Table 1 shows the average of the execution times for the proposed method and the conventional EVM, using a personal computer (Windows machine) with 3.33GHz processor and 3GB main memories. From the **Table 1**, one can see that the execution time of the ModEVM with mpinvl is the fastest compared with other EVMs. The reasons are that the ModEVM is carried out without calculating the eigenvectors of $\widetilde{\boldsymbol{R}}^{\dagger}\widetilde{\boldsymbol{B}}_i$ in (12) and the mpinvl has a property written in [12].

6. Conclusions

In this paper, by modifying the EVM, we have proposed an algorithm which can achieve the BD without calculating eigenvectors. Moreover, a combination of the modified EVM and the conventional EVM has been proposed. It can be seen that the combined EVM provides a better performance than the other EVMs in the case of $l = 1$, and the ModEVA with mpinvl provides a better performance than the other EVMs in the case of $l = 2$, but the average of execution time of the combined EVM is a little bit longer than the modified EVM. Although there exists such a trade-off, we conclude that our proposed EVM is more useful for solving the BD problem, because we consider that the performance accuracy is most important for achieving the BD.

Table 1. Comparison of the averages of the execution times.

Methods	times [sec] (l = 1)	times [sec] (l = 2)
The ModEVM with "pinv"	0.1429	0.1205
The ModEVM with "mpinvl"	0.1353	0.1168
The ModEVM with "mpinvl"	0.1492	0.1218
The ConEVM with "mpinvl"	0.1380	0.1179
The ComEVM with "mpinv"	0.1362	0.1175
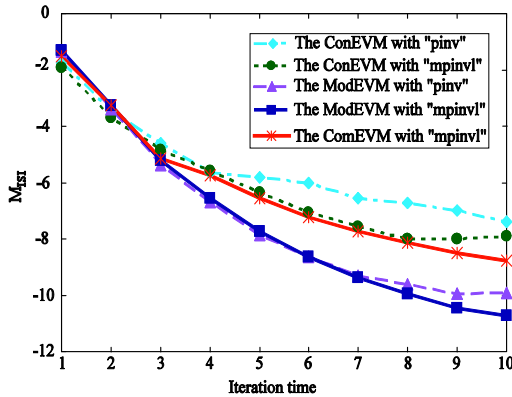
Figure 4. The performances of the proposed algorithm and the conventional methods (l = 2).

7. Acknowledgements

8. References

[1] Special issue on blind system identification and estimation, *Proceedings of the IEEE*, Vol. 86, No. 10, 1998, pp. 1907-2089.

[2] B. Jelonnek and K. D. Kammeyer, "A Closed-form Solution to Blind Equalization," *Signal Processing*, Vol. 36, No. 3, 1994, pp. 251-259. doi:10.1016/0165-1684 (94)90026-4

[3] M. Kawamoto, K. Kohno, Y. Inouye and K. Kurumatani, "A Modified Eigenvector Method for Blind Deconvolution of MIMO Systems Using the Matrix Pseudo-Inversion Lemma,", *International Symposium on Circuits and Systems* 2010, Paris, 30 May-2 June 2010, pp. 2514-2517.

[4] M. Kawamoto, K. Kohno and Y. Inouye, "Eigenvector Algorithms for Blind Deconvolution of MIMO-IIR Systems," *International Symposium on Circuits and Systems* 2007, New Orleans, 25-28 May 2007, pp. 3490-3493. doi: 10.1109/ISCAS.2007.378378

[5] M. Kawamoto, Y. Inouye and K. Kohno, "Recently Developed Approaches for Solving Blind Deconvolution of MIMO-IIR Systems: Super-Exponential and Eigenvector Methods," *International Symposium on Circuits and Systems* 2008, Seattle, 18-21 May 2008, pp. 121-124.

[6] B. Jelonnek, D. Boss and K. D. Kammeyer, "Generalized Eigenvector Algorithm for Blind Equalization," *Signal Processing*, Vol. 61, No. 3, 1997, pp. 237-264. doi:10.1016/ S0165-1684(97)00108-4

[7] M. Castella, *et al.*, "Quadratic Higher-Order Criteria for Iterative Blind Separation of a MIMO Convolutive Mixture of Sources," *IEEE Transactions. Signal Processing*, Vol. 55, No. 1, 2007, pp. 218-232. doi:10.1109/TSP.20 06.882113

[8] Y. Inouye and K. Tanebe, "Super-exponential Algorithms for Multichannel Blind Deconvolution," *IEEE Trans. Signal Processing*, Vol. 48, No. 3, 2000. pp. 881-888, doi: 10.1109/78.824685

[9] K. Kohno, Y. Inouye and M. Kawamoto, "Super Exponential Methods Incorporated with Higher-Order Correlations for Deflationary Blind Equalization of MIMO Linear Systems,"5*th International. Conference on Independent Component Analysis and Blind Signal Separation*, Granada, 22-24 September 2004, pp. 685-693.

[10] Shalvi and E. Weinstein, "Super-exponential methods for blind deconvolution," *The IEEE Transactions on Information Theory*, Vol. 39, No. 2, 1993, pp. 504-519. doi: 10.1109/18.212280

[11] 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 Transactions Circuits and Systems-I*, Vol. 55, No. 1, 2008, pp. 424-435. doi:10.1109/TCSI.2007.913613

[12] K. Kohno, M. Kawamoto and Y. Inouye, "A Matrix Pseudo-inversion Lemma and Its Application to Block-Based Adaptive Blind Deconvolution of MIMO Systems," *IEEE Transactions Circuits and Systems-I*, Vol. 57, No. 7, 2010, pp. 1449-1462. doi:10.1109/TCSI.2010. 2050222

[13] S. Haykin, "Adaptive Filter Theory," 3rd Edition, PrenticeHall, New Jersey, 1995.

CS

Appendix

The relationship between (13) and the SEM

The matrices $\widetilde{R}$ and $\widetilde{B}_i$ can be expressed as

$$\widetilde{R} = \widetilde{H}^H \widetilde{\Sigma} \widetilde{H} , \quad \widetilde{B}_i = \widetilde{H}^H \widetilde{\Lambda}_i \widetilde{H} \tag{28}$$

where $\widetilde{\Sigma}$ is a block-diagonal matrix which is denoted as $\tilde{\Sigma} := \text{block-diag}\{\Sigma_1,\cdots,\Sigma_n,\}$ $\Sigma := \text{diag}\{\cdots,\sigma_{s_i}^2,\sigma_{s_i}^2,\cdots,\}$ $i = \underline{1,n}$, $\widetilde{\Lambda}_i$ is a block-diagonal matrix which is represented as $\widetilde{\Lambda}_i := \text{block-diag}\{\Lambda_{i1},\cdots,\Lambda_{in}\}$,

$$\Lambda_{ij} := \text{diag}\left\{\cdots,\left|g_{ij}(-1)\right|^2 \gamma_j,\left|g_{ij}(0)\right|^2 \gamma_j,\cdots,\right\} \tag{29}$$

$j = \underline{1,n}$. Then, from (5) and (28), $\widetilde{d}_i = \widetilde{B}_i \widetilde{w}_i$ can be expressed as

$$\widetilde{d}_i = \widetilde{B}_i \widetilde{w}_i = \widetilde{H}^H \widetilde{\Lambda}_i \widetilde{g}_i. \tag{30}$$

It can be seen from (29) that the elements of $\widetilde{\Lambda}_i \widetilde{g}_i$ are $\left|g_{ij}(k)\right|^2 g_{ij}(k)\gamma_j$, $k = -\underline{\infty,\infty}$. Here, we define the following equation:

$$f_{ij}(k) = \frac{\gamma_j}{\sigma_{s_j}^2}\left|g_{ij}(k)\right|^2 g_{ij}(k). \tag{31}$$

This can be used for the SEM with respect to $g_{ij}(k)$, using the 4th order cumulant. [7]   Substituting (31) into (30), we obtain the following equation:

$$\widetilde{d}_i = \widetilde{H}^H \widetilde{\Sigma} \widetilde{f}_i, \tag{32}$$

where

$$\tilde{f}_i = \left[f_{i1}^T, f_{i2}^T,\cdots, f_{in}^T\right]^T = \left[\cdots, f_{ij}(1), f_{ij}(0), f_{ij}(1),\cdots\right]^T.$$

Moreover, substituting (28) and (32) into (13), then (13) can be expressed as

$$\tilde{w}_i = \left(\tilde{H}^H \tilde{\Sigma} \tilde{H}\right)^\dagger \tilde{H}^H \tilde{\Sigma} \tilde{f}_i, \tag{33}$$

where λ_i is assumed to be 1. (33) is the first step of the SEM with respect to $\widetilde{w}_i$ [8]. In the SEM, the second step, that is, the normalization step is implemented using (14). Therefore, (13) and (14) are nothing but the iterative algorithm of the SEM. This completes the proof.