SUPER-EXPONENTIAL METHODS FOR MULTICHANNEL BLIND DECONVOLUTION

Yujiro Inouye

Department of Electronic and Control
Systems Engineering, Shimane University,
Matsue, Shimane 690, Japan
inouye@riko.shimane-u.ac.jp

Kazuaki Tanebe

Department of Systems Engineering,
Osaka University,
Toyonaka, Osaka 560, Japan
tanebe@hopf.sys.es.osaka-u.ac.jp

ABSTRACT

Multichannel blind deconvolution has received increasing attention during the last decade. Recently, Martone [3, 4] extended the super-exponential method proposed by Shalvi and Weinstein [1, 2] for single-channel blind deconvolution to multichannel blind deconvolution. However, the Martone extension suffers from two type of serious drawbacks. The objective of this paper is to obviate these drawbacks and to propose three approaches to multichannel blind deconvolution. In the first approach, we present a multichannel super-exponential algorithm. In the second approach, we present a super-exponential deflation algorithm. In the third approach, we present a two-stage super-exponential algorithm.

1. INTRODUCTION

Blind deconvolution, in particular, multichannel blind deconvolution has received increasing attention during the last decade, and it arises in a wide variety applications, for example, in array processing for wideband sources under multipath propagation, in speech and image enhancement, and in digital communication; see [1]-[5] and references therein.

Recently, Shalvi and Weinstein proposed an attractive approach to single-channel blind deconvolution (which is called the *super-exponential method* [1, 2]), which was extended to the multichannel case by Martone [3, 4]. However, the Martone algorithm suffers from two types of serious drawbacks. The first one is that it can be only applied to the case where all the elements of input sources possess identical variances and identical fourth-order cumulants. The second one is that the algorithm fails to converge to a desired solution, except when it starts in a small vicinity of the desired solution (this means the local convergence of the algorithm is guaranteed). The objective of this paper is to obviate these drawbacks and to derive super-exponential algorithms for multichannel blind deconvolution.

In this paper, we propose three approaches to multichannel blind deconvolution. In the first approach, we present a multichannel super-exponential algorithm, which can be applied to a general case where the elements of input signals possess various variances and various fourth-order cumulants. This algorithm is an extension of the Martone algorithm to the case where the elements have various variances and various fourth-order cumulants. In the second approach, we present a super-exponential deflation algorithm. This algorithm is input-iterative, i.e., the input signals are extracted at each output and cancelled from each output one by one. Thus the number of the input signals (sources) is deflated one by one in the algorithm. In the third approach, we present a two-stage super-exponential algorithm, which unifies the above two algorithms. It is a two-stage algorithm. In the first stage, the first algorithm is used to obtain rough results for the initialization of the second algorithm. Then the second algorithm is employed to obtain better results. Simulation examples are presented to illustrate the performance of the proposed algorithms.

The following notation will be used in this paper. The superscripts $*$, T and $\dagger$ denote respectively the complex conjugate, the transpose and the pseudo-inverse operations of a matrix. The (i,j)th element of a matrix X is denoted by $x_{i,j}$, and the ith element of a vector x is denoted by x_i. The variance or the second-order cumulant of random variable x is denoted by σ_x^2 or $\text{cum}_2(x)$. Its fourth-order cumulant is denoted by $\text{cum}_4(x)$. The joint cumulant of random variables $x_1, x_2, \cdots, x_n$ is denoted by $\text{cum}\{x_1, x_2, \cdots, x_n\}$.

2. PROBLEM FORMULATION

Let us consider the cascade system of an unknown LTI system with n inputs and m outputs and an equalizer with m inputs and n outputs, which is illustrated in Fig. 1.

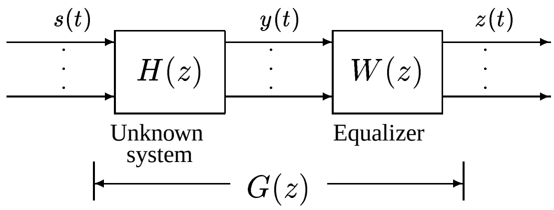

Figure 1: Cascade system of the Unknown System and an Equalizer

We make the following assumptions on the systems and signals involved

(A1) The system $H(z)$ is unknown. It is stable in the sense that

$$\sum_{k=-\infty}^{\infty} \|H(k)\| < \infty, \tag{1}$$

where $\|\cdot\|$ denotes the Euclidean matrix norm.

(A2) The transfer function $H(z)$ of the unknown system is of 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-inverse of the unknown system which is stable in the sense of (1)).

(A3) The input sequence $\{\boldsymbol{s}(t)\}$ is a zero-mean , non-Gaussian random vector process, whose element process $\{s_i(t)\}$, $i = 1, 2, \cdots, n$ are mutually independent. Moreover, each element process $\{s_i(t)\}$ is an independently and identically distributed (i.i.d) process with nonzero variance $\sigma_i^2 \neq 0$ and nonzero fourth-order cumulant $\gamma_i \neq 0$.

(A4) The equalizer $W(z)$ is stable. It is assumed to be an FIR system of sufficient length L, so that the truncation effect can be ignored.

In a vector form, the cascade system $G(z)$ can be written as

$$\tilde{\boldsymbol{g}}_i = \tilde{\boldsymbol{H}}\tilde{\boldsymbol{w}}_i, \quad i = 1, 2, \cdots, n, \tag{2}$$

where $\tilde{g}_i$ is the column vector (of possibly infinite elements) consisting of the ith-output impulse responses of the cascade system defined by

$$\tilde{\boldsymbol{g}}_i := [\boldsymbol{g}_{i,1}^T, \boldsymbol{g}_{i,2}^T, \cdots, \boldsymbol{g}_{i,n}^T]^T, \tag{3}$$

$$\boldsymbol{g}_{i,j} := [\cdots, g_{i,j}(-1), g_{i,j}(0), g_{i,j}(1), \cdots]^T, \tag{4}$$

$\tilde{w}_i$ is the Lm-column vector consisting of the tap coefficients (corresponding to the i-th output) of the equalizer defined by

$$\tilde{\boldsymbol{w}}_i := [\boldsymbol{w}_{i,1}^T, \boldsymbol{w}_{i,2}^T, \cdots, \boldsymbol{w}_{i,m}^T]^T, \tag{5}$$

$$\boldsymbol{w}_{i,j} := [w_{i,j}(L_1), w_{i,j}(L_1+1), \cdots, w_{i,j}(L_2)]^T, \tag{6}$$

and $\tilde{H}$ is the $n \times m$ block matrix defined by

$$\tilde{\boldsymbol{H}} := \begin{bmatrix} \boldsymbol{H}_{1,1} & \boldsymbol{H}_{1,2} & \cdots & \boldsymbol{H}_{1,m} \\ \boldsymbol{H}_{2,1} & \boldsymbol{H}_{2,2} & \cdots & \boldsymbol{H}_{2,m} \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{H}_{n,1} & \boldsymbol{H}_{n,2} & \cdots & \boldsymbol{H}_{n,m} \end{bmatrix}, \tag{7}$$

whose (i,j)th block element $H_{i,j}$ is the matrix (of L columns and possibly infinite number of rows) with the (l,r)th element $[H_{i,j}]_{l,r}$ defined by

$$[\boldsymbol{H}_{i,j}]_{l,r} := h_{j,i}(l-r), \quad -\infty < l < +\infty, L_1 \leq r \leq L_2. \tag{8}$$

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

$$[\tilde{g}_1, \cdots, \tilde{g}_n] = \tilde{H}[\tilde{w}_1, \cdots, \tilde{w}_n] = [\tilde{\delta}_1, \cdots, \tilde{\delta}_n]P, \tag{9}$$

where P is an $n \times n$ permutation matrix and $\tilde{\delta}_i$ is the n-block column vector defined by

$$\tilde{\boldsymbol{\delta}}_i := [\delta_{i,1}^T, \delta_{i,2}^T, \cdots, \delta_{i,n}^T]^T \tag{10}$$

$$\delta_{i,j} := \begin{cases} \hat{\delta}_i & \text{if } i = j \\ (\cdots, 0, 0, 0, \cdots)^T & \text{otherwise} \end{cases} \tag{11}$$

Here $\hat{\delta}_i$ is the column vector (of infinite elements) whose rth element $\hat{\delta}_i(r)$ is given by

$$\hat{\delta}_i(r) = d_i \delta(r - k_i) \tag{12}$$

where $\delta(t)$ is the Kronecker delta function, d_i is a complex number standing for both the scale change and the phase shift, and k_i is an integer standing for the time shift.

3. MULTICHANNEL SUPER-EXPONENTIAL ALGORITHM

In this section, we extend the Martone algorithm [3] for a restricted case to a general case where the elements of input signals $s_i(t)$'s possess various variances and various fourth-order cumulants. By adjusting the elements $g_{i,j}(k)$'s for the cascade system, it is possible to obtain an iterative procedure that converges to a desired solution.

Martone [3] proposed the following two-step iterative procedure:

$$g_{i,j}(k)^{[1]} = (g_{i,j}(k))^p (g_{i,j}^*(k))^q, \tag{13}$$

$$g_{i,j}(k)^{[2]} = g_{i,j}(k)^{[1]} \frac{1}{\sqrt{\sum_{j=1}^n \sum_l |g_{i,j}(l)^{[1]}|^2}}. \tag{14}$$

where $(\cdot)^{[1]}, (\cdot)^{[2]}$ stand for the result of the first step and the result of the second step for an iteration, and p and q are positive integers such that $p + q \geq 2$. In this paper, we consider the following two-step procedure:

$$g_{i,j}(k)^{[1]} = \frac{\rho_j}{\sigma_j^2} (g_{i,j}(k))^p (g_{i,j}^*(k))^q, \tag{15}$$

$$g_{i,j}(k)^{[2]} = g_{i,j}(k)^{[1]} \frac{1}{\sqrt{\sum_{j=1}^n \sum_l \sigma_j^2 |g_{i,j}(l)^{[1]}|^2}}. \tag{16}$$

where

$$\rho_j = \mathrm{cum}(\underbrace{s_j(t), \cdots, s_j(t)}_{p}, \underbrace{s_j^*(t), \cdots, s_j^*(t)}_{q+1})$$

and $\sigma_j^2 = \mathrm{cum}_2(s_j(t))$ for $j = 1, 2, \cdots, n$.

In this paper, for notationally simplicity, we confine ourselves to the case $p = 2$ and $q = 1$ (which gives a solution in terms of fourth-order cumulants), although our results are expandable to a general (p, q) case (higher-order cumulant case).

We turn to the two-step procedure (15) and (16) with $p = 2$, $q = 1$ and $\rho_j = \gamma_j$ for $j = 1, \cdots, n$. According to (15), let us define

$$f_{i,j}(k) := \frac{\gamma_j}{\sigma_j^2} g_{i,j}^2(k) g_{i,j}^*(k) \tag{17}$$

and put

$$\tilde{\boldsymbol{f}}_i := [\boldsymbol{f}_{i,1}^T, \boldsymbol{f}_{i,2}^T, \cdots, \boldsymbol{f}_{i,n}^T]^T, \tag{18}$$

$$\boldsymbol{f}_{i,j} := [\cdots, f_{i,j}(-1), f_{i,j}(0), f_{i,j}(1), \cdots]^T. \tag{19}$$

Taking account of (9), we want to find equalizer tap vectors $\tilde{w}_i$'s by solving the weighted least squares problem

$$\min_{\tilde{w}_i} (\tilde{H}\tilde{w}_i - \tilde{f}_i)^{T*} \tilde{\Sigma} (\tilde{H}\tilde{w}_i - \tilde{f}_i), \quad i = 1, 2, \cdots, n, \tag{20}$$

whose solutions denoted by $\tilde{w}_i^{[1]}$'s are easily given by

$$\tilde{w}_i^{[1]} = (\tilde{H}^{T*}\tilde{\Sigma}\tilde{H})^{-1}\tilde{H}^{T*}\tilde{\Sigma}\tilde{f}_i, \quad i = 1, 2, \cdots, n. \tag{21}$$

According to the weighted normalization in (16), it is easily shown that the second step is reduced to

$$\tilde{\boldsymbol{w}}_i^{[2]} = \frac{\tilde{\boldsymbol{w}}_i^{[1]}}{\sqrt{\tilde{\boldsymbol{w}}_i^{[1]T*}(\tilde{\boldsymbol{H}}^{T*}\tilde{\Sigma}\tilde{\boldsymbol{H}})\tilde{\boldsymbol{w}}_i^{[1]}}}, \quad i = 1, 2, \cdots, n, \tag{22}$$

where $\tilde{\Sigma}$ is the $n \times n$ block diagonal matrix defined by

$$\tilde{\Sigma} := \begin{bmatrix} \Sigma_{1,1} & 0 & \cdots & 0 \\ 0 & \Sigma_{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Sigma_{n,n} \end{bmatrix}, \qquad (23)$$

$$\Sigma_{i,i} := \sigma_i^2 I, \quad i = 1, 2, \cdots, n.$$

Here I denotes the identity matrix of possibly infinite number of columns and rows.

Then it will be shown that the two steps (21) and (22) can be expressed in terms of the variances and covariances of the outputs of the original system and of the fourth-order (cross-) cumulants of the system outputs and the equalizer outputs. By straight forward calculations, we obtain the following two steps of each iteration of the multichannel super-exponential algorithm:

$$\tilde{\boldsymbol{w}}_i^{[1]} = \tilde{\boldsymbol{R}}^{-1} \tilde{\boldsymbol{D}}_i, \quad i = 1, 2, \cdots, n, \qquad (24)$$

$$\tilde{\boldsymbol{w}}_i^{[2]} = \frac{\tilde{\boldsymbol{w}}_i^{[1]}}{\sqrt{\tilde{\boldsymbol{w}}_i^{[1]T*} \tilde{\boldsymbol{R}} \tilde{\boldsymbol{w}}_i^{[1]}}}, \quad i = 1, 2, \cdots, n, \qquad (25)$$

where $\tilde{\boldsymbol{R}}$ is the $m \times m$ block matrix defined by

$$\tilde{R} = \begin{bmatrix} \boldsymbol{R}_{1,1} & \boldsymbol{R}_{1,2} & \cdots & \boldsymbol{R}_{1,m} \\ \boldsymbol{R}_{2,1} & \boldsymbol{R}_{2,2} & \cdots & \boldsymbol{R}_{2,m} \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{R}_{m,1} & \boldsymbol{R}_{m,2} & \cdots & \boldsymbol{R}_{m,m} \end{bmatrix} \qquad (26)$$

whose (i, j)th block element $\boldsymbol{R}_{i,j}$ is the $L \times L$ matrix with the (l, r)th element $[R_{i,j}]_{l,r}$ defined by

$$[\boldsymbol{R}_{i,j}]_{l,r} := \mathrm{cum}(y_j(t-r), y_i^*(t-l)), \qquad (27)$$

and $\boldsymbol{D}_i$ is the n-block vector defined by

$$\tilde{\boldsymbol{D}}_i := \left[\boldsymbol{d}_{i,1}^T, \boldsymbol{d}_{i,2}^T, \cdots, \boldsymbol{d}_{i,n}^T\right]^T \qquad (28)$$

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

$$[\boldsymbol{d}_{i,j}]_r := \mathrm{cum}(z_i(t), z_i(t), z_i^*(t), y_j^*(t-r)). \qquad (29)$$

4. SUPER-EXPONENTIAL DEFLATION ALGORITHM

In this section, we present a deflation algorithm which converges globally to a desired solution except for pathological cases (for example, there are two leading taps of the initial taps of the cascade system, that exactly take the same absolute value). This algorithm is input-iterative, i.e., the input signals are extracted at each output and cancelled at each output one by one. In this algorithm, we first apply the multichannel super-exponential algorithm presented in Section 3 to the outputs of the original system with m outputs and n inputs to extract only one input, and we estimate the contribution of the extracted input to the outputs. Then we remove this contribution from the outputs to define the output of a multichannel system with m outputs and $n-1$ inputs. We next apply the multichannel super-exponential algorithm to the outputs of the system with m outputs and $n-1$ inputs to extract the next input. We continue this process successively until we extract the last input. Now let us introduce n mL-column vectors as intermediate tap-coefficient vectors of equalizers as follows:

$$\tilde{\boldsymbol{c}}_i := [\boldsymbol{c}_{i,1}^T, \boldsymbol{c}_{i,2}^T, \cdots, \boldsymbol{c}_{i,m}^T]^T, \quad i = 1, \cdots, n \qquad (30)$$

$$\boldsymbol{c}_{i,j} := [c_{i,j}(L_1), c_{i,j}(L_1+1), \cdots, c_{i,j}(L_2)]^T. \qquad (31)$$

Then the n input signals are extracted successively by using the following algorithm.

Super-Exponential Deflation Algorithm

Step 1. Set $i = 1$ (where i denote the order of an input extracted).

Step 2. Carry out the following iterations enough to extract an input: Each of the iterations consist of the two steps as follows.

$$\tilde{\boldsymbol{c}}_i^{[1]} = \tilde{\boldsymbol{R}}_i^{\dagger} \boldsymbol{D}_i \qquad (32)$$

$$\tilde{\boldsymbol{c}}_i^{[2]} = \frac{\tilde{\boldsymbol{c}}_i^{[1]}}{\sqrt{\tilde{\boldsymbol{c}}_i^{[1]T*} \tilde{\boldsymbol{R}}_i \tilde{\boldsymbol{c}}_i^{[1]}}}. \qquad (33)$$

where $\tilde{R}_i$ and $\tilde{D}_i$ are respectively calculated by (26) along with (27) and (28) along with (29) using the values of the outputs $y_k(t)$'s $(k = 1, \cdots, m)$ and the values of the equalizer outputs $z_i(t)$'s with $w_{i,j}(k)$'s replaced by the corresponding values of $c_{i,j}(k)$'s obtained before the iteration.

Step 3. As a possibly scaled and time-shifted estimate of an input $s_{j_i}(t)$, calculate the equalizer output $z_i(t)$ by

$$z_i(t) = \sum_{j=1}^{m} \sum_{k} c_{i,j}(k) y_j(t-k), \qquad (34)$$

where $c_{i,j}(k)$'s are the new values obtained in Step 2. Then cross-correlate $z_i(t)$'s with the outputs $y_k(t)$'s and define a possibly scaled and time-shifted estimate of $h_{k,j_i}(\tau)$ as

$$\hat{h}_{k,j_i}(\tau) := \mathrm{E}\{y_k(t) z_i^*(t-\tau)\}, \quad k = 1, 2, \cdots, m. \qquad (35)$$

Then consider the reconstructed contribution of $z_i(t)$ to the outputs $y_k(t)$'s, defined by

$$\hat{y}_{k,j_i}(t) := \sum_{\tau} \hat{h}_{k,j_i}(\tau) z_i(t-\tau). \qquad (36)$$

Step 4. Remove the above contribution from the outputs $y_k(t)$'s to define the outputs of a linear system with m outputs and $n-1$ inputs. These are given by

$$y_k^{(i)}(t) := y_k(t) - \hat{y}_{k,j_i}(t), \quad k = 1, \cdots, m \qquad (37)$$

Step 5. If $i < n$, then set $i = i+1$ and $y_k(t) = y_k^{(i)}(t)$ for $k = 1, \cdots, m$, and go back to Step 2. If $i = n$, then stop here.

In implementing the above algorithm, all the cumulants in (27) and (29) and all the expectations in (35) are replaced with their samples averages over appropriate data records.

Table 1: The average $\mathrm{M}_{\mathrm{ISI}}$ values over 50 Monte Calro runs

	$\mathrm{M}_{\mathrm{ISI}}$(dB)
Super-Exponential Algorithm	9.8889
Super-Exponential Deflation Algorithm	-18.0110
Two-Stage Super-Exponential Algorithm	-21.0904

5. TWO-STAGE SUPER-EXPONENTIAL ALGORITHM

The multichannel super-exponential algorithm of Section 3 converges to a desired solution only if it starts in a small neighborhood of the desired solution, while the super-exponential deflation algorithm of Section 4 converges to a desired solution almost always (i.e., except for pathological cases). However, we have experienced about the second algorithm through simulation experiments that the results of recovered input signals gradually degrade as the order of the recovered input signals increases. In order to remedy this defect, we propose a two-stage super-exponential algorithm as follows.

Two-Stage Super-Exponential Algorithm

Stage 1. Use the super-exponential deflation algorithm to find rough results on tap-coefficients $c_{i,j}(k)$'s and impulse-response estimates $\hat{h}_{k,j_i}(\tau)$'s as the initialization required in Stage 2.

Stage 2. Use the multichannel super-exponential algorithm to find better results on over-all equalizer $W(z)$.

6. SIMULATIONS

In order to see the performance of the proposed algorithms, we considered the following example. The unknown system was given by

$$\boldsymbol{H}(z) = \begin{bmatrix} \frac{0.5+z^{-1}}{1+0.5z^{-1}} & 0 \\ 0 & \frac{0.2+z^{-1}}{1+0.2z^{-1}} \end{bmatrix} \begin{bmatrix} \frac{1}{2} & \frac{-\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix}$$

and the first component of input signals $\{s_1(t)\}$ was taken from a 16-QAM source and the second component of input signals $\{s_2(t)\}$ was taken from a 4-QAM source. As a measure of performance we used the multichannel intersymbol interference denoted by $\mathrm{M}_{\mathrm{ISI}}$, defined by

$$\begin{aligned} \mathrm{M}_{\mathrm{ISI}} \quad := \quad & \sum_{i=1}^{n} \frac{|\sum_{j=1}^{n}\sum_{t=-\infty}^{\infty}|g_{i,j}(t)|^2 - |g_{i,\cdot}|^2_{\max}|}{|g_{i,\cdot}(\cdot)|^2_{\max}} \\ + \quad & \sum_{j=1}^{n} \frac{|\sum_{i=1}^{n}\sum_{t=-\infty}^{\infty}|g_{i,j}(t)|^2 - |g_{\cdot,j}|^2_{\max}|}{|g_{\cdot,j}(\cdot)|^2_{\max}} \end{aligned}$$

The length L of the equalizers was chosen to be 12. First, we set the values of the tap coefficients $\{w_{i,j}(t) : t = 0, 1, \cdots, 11; i, j = 1, 2\}$ to be zero, except for $w_{1,1}(5) = w_{2,2}(5) = 2/3$ and $w_{1,2}(5) = w_{2,1}(5) = 1/3$, and then we examined the first algorithm using these values as the initialization of the equalizer. The algorithm failed always to converge to a desired solution, and failed to recover the second source. We next examined the second algorithm using randomly chosen values as the initialization of the intermediate equalizers, and then examined the third algorithm using the results obtained by the second algorithm as the initialization of the over-all equalizer. The second and the third algorithms converged to a desired solution, and succeeded in recovering the first and the second sources. The three algorithms were respectively tested in 50 independent Monte Carlo runs using 3,000 data samples of each of the two outputs. The average $\mathrm{M}_{\mathrm{ISI}}$ is shown in Table 1 for each algorithm. It can be seen from Table 1 that the third algorithm shows better performance than the second algorithm.

7. CONCLUSIONS

The Martone algorithm for multichannel blind deconvolution undergoes two types of serious drawbacks. To remedy these drawbacks, we proposed the three approaches to multichannel blind deconvolution. In the first approach, we presented the multichannel super-exponential algorithm which is locally convergent but can be applied to the general case where the elements of input signals possess various variances and various fourth-order cumulants. In the second approach, we presented the super-exponential deflation algorithm which is globally convergent almost always but has worse performance than the first algorithm. In the third approach, we proposed the two-stage super-exponential algorithm which unifies the advantages of the above two algorithms. Simulation examples illustrated the performance of the proposed algorithms.

REFERENCES

[1] O. Shalvi and E. Weinstein, “Super-exponential methods for blind deconvolution,” *IEEE Trans. Inform. Theory*, vol. 39, pp. 504-519, 1993.

[2] O. Shalvi and E. Weinstein, “Universal methods for blind deconvolution,” *Blind Deconvolution*, ed. S. Haykin, New Jersey: Prentice-Hall, pp. 121-180, 1994.

[3] M. Martone, “Blind multichannel deconvolution in multiple access spread spectrum communications using higher order statistics,”*IEEE Proc. of ICC '95*, pp. 49-53, 1995.

[4] M. Martone, “Adaptive antenna array system for wireless communication systems using spatial-temporal higher order statistics filtering,” *Proc. ICC '95*, pp. 1505-1508, 1995.

[5] J. K. Tugnait, “Identification and deconvolution of multicannel linear non-Gaussian processes using higher order statistics and inverse filter criteria,”*IEEE Trans. on Signal Processing*, vol. 45, pp. 658-672, 1997.