BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES
Vol. 53, No. 1, 2005

A two-step approach to blind deconvolution of speech and sound sources in the time domain

F.A. OKAZAKI* and W. KASPRZAK**

Institute of Control and Computation Engineering, Warsaw University of Technology,
15/19 Nowowiejska St., 00-665 Warszawa, Poland

Abstract. In order to understand commands given through voice by an operator, user or any human, a robot needs to focus on a single source, to acquire a clear speech sample and to recognize it. A two-step approach to the deconvolution of speech and sound mixtures in the time-domain is proposed. At first, we apply a deconvolution procedure, constrained in the sense, that the de-mixing matrix has fixed diagonal values without non-zero delay parameters. We derive an adaptive rule for the modification of the de-convolution matrix. Hence, the individual outputs extracted in the first step are eventually still self-convolved. This corruption we try to eliminate by a de-correlation process independently for every individual output channel.

Key words: blind signal analysis, convolved mixtures, independent component analysis, robotic sensors, speech reconstruction.

1. Introduction

An autonomous service robot requires the use of many types of sensors [1,2]. Some of these sensors can be microphones that record sound and speech, originated in robot's environment. Especially, in order to understand commands given through voice by an operator, user or any human, the robot needs to focus on a single source, to acquire a clear speech sample and to recognize it. The focussing and source reconstruction steps can be modelled in terms of blind signal deconvolution, based on blind signal processing [3] and independent component analysis [4].

Assuming that several sensors capture mutually convoluted signals of several speech and sound sources the goal of blind processing is to reconstruct the sources (i.e. their waveforms) without any prior knowledge about the sources and the convolution process.

Two types of approaches can be distinguished for such a task: a multi-channel blind source deconvolution (MBD), performed in the time-domain [5,6], or a multiple blind source separation (BSS), performed in the frequency domain (one separation process for every single frequency bin) [7,8]. In both cases without some constraints, put onto the sources or the convolutive mixing process, a general solution is difficult if not impossible to achieve.

In the past, the authors of this paper considered general approaches of both types. These approaches worked well if some narrow conditions have been satisfied: a general deconvolution approach in the time-domain required a perfect knowledge about the auto-correlation structure of each source signal [6], whereas the frequency-based approach required the existence of a dominating frequency bin component and a prior-knowledge about it [9]. These assumptions can hardly be satisfied by sound and speech signals. This fact is already well recognized in the community, hence approaches are considered that profit from useful constraints put onto the sources and mixing process [10,11].

In our paper a two-step approach to the deconvolution of speech and sound mixtures in the time-domain is proposed. At first, we apply a deconvolution procedure, constrained in the sense, that the de-mixing matrix has fixed diagonal values without non-zero delay parameters. This is a generalization of the de-correlation approach, presented by [12]. We derive an adaptive rule for the modification of the de-convolution matrix. Hence, the individual outputs extracted in the first step are eventually self-convolved. We try to eliminate it by a de-correlation process, performed independently for every individual output channel, like applied for the task of convoluted noise elimination [13].

We start in the second section with the definition of the blind source deconvolution (MBD) problem and with the explanation of our proposed approach – the ECDA method for MBD and an adaptive rule for single-channel equalization. In section 3 we describe the derivation of the update rule for the ECDA method. The consecutive section contains some experimental results. We conclude the work by the summary section.

2. The two-step approach to MBD

2.1. The problem. Let us first explain the problem of source deconvolution on the base of some image sources. In Figure 1 three convoluted image mixtures of 3 different sources are shown. If we know exactly the autocorrelation structure of all three sources then a one-step

*e-mail: a.okazaki@elka.pw.edu.pl
**e-mail: w.kasprzak@ia.pw.edu.pl

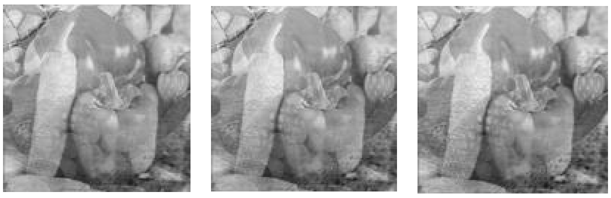

Fig. 1. Three convolution mixtures of 3 images

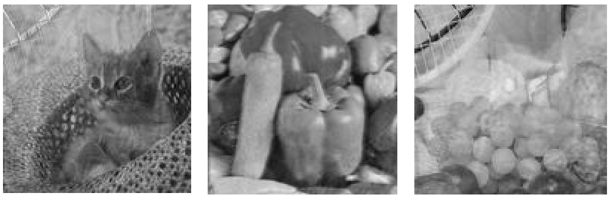

Fig. 2. Three estimated sources from the mixtures in Fig. 1 if the auto-correlation structures of sources are a priori known (WWW of source images: www.bip.riken.go.jp)

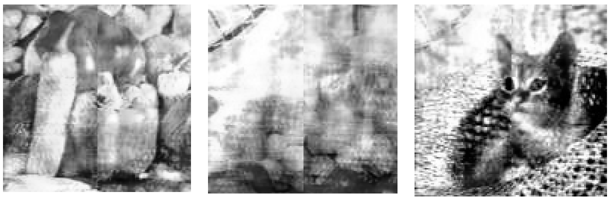

Fig. 3. A one-step deconvolution is not fully possible if the auto-correlation structures of sources are not known

de-mixing process provides already high-quality estimations of unknown mixing matrices and unknown sources (Fig. 2). But it is rather unusual to have this specific knowledge in practice. Then our results will be like shown in Fig. 3. The goal of this paper is to split the deconvolution process into two consecutive steps. In the first step the between-source mixtures are blindly deconvolved, whereas in the second step a blind decorrelation of individual channels with auto-correlated sources is performed.

2.2. MBD under constant mixing matrix diagonal. About the unknown mixing of n unknown sources $\{s_i(k)(i = 1, \ldots, n)\}$ we assume that the sources are statistically independent and their mixtures $\{x_i(k)(i = 1, \ldots, n)\}$ are (discrete- and finite-time) convolutions of $\{s_i(k)\}$.

For a single measurement (input) channel with index i the measured mixture signal is [2]:

$$x_i(k) = \sum_{j=1}^{n} \sum_{l=0}^{L} h_{ij}(l) s_j(k-l) \qquad (1)$$

where L means the order of the FIR filter (the number of time delays) and $[h_{ij}]$ means mixing coefficient vectors. The mixing coefficients for the j-th source in the i-th input channel are given by the $(p+1)$-elementary vector:

$$\boldsymbol{h}_{ij}^T = [h_{ij}(0), \ \ldots \ , h_{ij}(p)] \, .$$

The deconvolution of input mixtures results in outputs:

$$y_i(k) = \sum_{j=1}^{n} \sum_{b=0}^{q} w_{ij}(b) \times x_j(k-b), \ i = 1, \ldots, n.$$

To achieve deconvolution the unknown coefficients

$$\boldsymbol{w}_{ij}^T = [w_{ij}(0), \ \ldots \ , w_{ij}(q)] \, . \qquad (2)$$

must be estimated, for every j-th input and i-th output. The matrix form of the mixing and demixing equations is as follows:

$$\begin{pmatrix} x_1(k) \\ \vdots \\ x_n(k) \end{pmatrix} = \begin{pmatrix} h_{11}^T & \cdots & h_{1n}^T \\ \vdots & \ddots & \vdots \\ h_{n1}^T & \cdots & h_{nn}^T \end{pmatrix} \begin{pmatrix} s_1(k) \\ \vdots \\ s_n(k) \end{pmatrix} \qquad (3a)$$

$$\begin{pmatrix} y_1(k) \\ \vdots \\ y_n(k) \end{pmatrix} = \begin{pmatrix} w_{11}^T & \cdots & w_{1n}^T \\ \vdots & \ddots & \vdots \\ w_{n1}^T & \cdots & w_{nn}^T \end{pmatrix} \begin{pmatrix} x_1(k) \\ \vdots \\ x_n(k) \end{pmatrix} \qquad (3b)$$

Please note, that all matrix or vector elements on the right hand side of above equation (3) are itself vectors – in accordance with considered delay entities. For example:

$$s_i^T(k) = [s_i(k), \ \ldots, \ s_i(k-p)] \qquad (4a)$$

$$x_i^T(k) = [x_i(k), \ \ldots, \ x_i(k-q)] \qquad (4b)$$

An equivalent representation in the Z-domain is:

$$\boldsymbol{F}(z) = \sum_{k=0}^{K} f(k) z^{-k}, \quad \boldsymbol{X}_i(z) = \sum_{j=1}^{n} Hij(z) S_j(z)$$

$$\boldsymbol{X}(z) = \boldsymbol{H}(z) \boldsymbol{S}(z) \qquad (5)$$

where

$$\boldsymbol{H}(z) = \begin{pmatrix} H_{11}(z) & \cdots & H_{1n}(z) \\ \vdots & \ddots & \vdots \\ H_{n1}(z) & \cdots & H_{nn}(z) \end{pmatrix},$$

$$\boldsymbol{x}(z) = \begin{pmatrix} x_1(z) \\ \vdots \\ x_n(z) \end{pmatrix}, \quad \boldsymbol{s}(z) = \begin{pmatrix} s_1(z) \\ \vdots \\ s_n(z) \end{pmatrix}. \qquad (6)$$

The deconvolution problem can be simplified by assuming a specific mixing matrix. In fact let us fix the diagonal coefficients of this matrix, following the proposition in [8]:

$$\mathbf{H}(z) = \mathbf{H}_a(z) \, \mathbf{H}_b(z) \qquad (7)$$

$$\boldsymbol{H}_a(z) = \begin{pmatrix} 1 & \frac{H_{21}(z)}{H_{11}(z)} & \cdots & \frac{H_{1n}(z)}{H_{nn}(z)} \\ \frac{H_{21}(z)}{H_{11}(z)} & \ddots & & \vdots \\ \vdots & & \ddots & \frac{H_{(n-1)n}(z)}{H_{nn}(z)} \\ \frac{H_{n1}(z)}{H_{11}(z)} & \cdots & \frac{H_{n(n-1)}(z)}{H_{(n-1)(n-1)}(z)} & 1 \end{pmatrix},$$

$$\boldsymbol{H}_b(z) = \begin{pmatrix} H_{11}(z) & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & H_{nn}(z) \end{pmatrix} \qquad (8)$$

Hence

$$\mathbf{X}(z) = \mathbf{H}(z)\mathbf{S}(z) = \mathbf{H}_a(z)\mathbf{H}_b(z)\mathbf{S}(z) = \mathbf{H}_a(z)\mathbf{S}'(z) \quad (9)$$

where

$$\boldsymbol{S}'(z) = \begin{pmatrix} s_i{}'(z) \\ \vdots \\ s_n{}'(z) \end{pmatrix} = \begin{pmatrix} H_{11}(z)s_1(z) \\ \vdots \\ H_{nn}(z)s_n(z) \end{pmatrix}. \quad (10)$$

This assumption means in practice that every source is recorded in one channel without any delay – only one source per input exhibits these characteristics. This is a simplification that can be justified in practice, when every microphone is located near to a single source only and far to all other sources.

The demixing process seeks for a separation matrix representing an FIR filter:

$$\mathbf{W}(z) = \mathbf{P}(z)\mathrm{cof}\mathbf{H}(z) \quad (11a)$$

under the condition of properly defined mixing, i.e. it should hold $(\det(\mathbf{H}(z)) \neq 0)$. $\mathrm{cof}(\mathbf{H}(z))$ is related to the inverse matrix $\mathbf{H}^{-1}(z)$ in such a way that:

$$\mathrm{cof}(\mathbf{H}(z)) = \det(\mathbf{H}(z))\mathbf{H}^{-1}(z). \quad (11b)$$

The matrix $\mathbf{P}(z)$ is a permutation matrix. Hence:

$$\begin{aligned} \mathbf{Y}(z) &= \mathbf{W}(x)\mathbf{X}(z) = \mathbf{W}(z)\mathbf{H}(z)\mathbf{S}(z) \\ &= \mathbf{P}(z)\mathrm{cof}(\mathbf{H}(z))\mathbf{H}(z)\mathbf{S}(z) \\ &= \mathbf{P}(z)\det(\mathbf{H}(z))\mathbf{H}^{-1}(z)\mathbf{H}(z)\mathbf{S}(z) \\ &= \mathbf{P}(z)\det(\mathbf{H}(z))\mathbf{S}(z) \end{aligned} \quad (12)$$

As the goal of the de-mixing process is to achieve statistically independent output signals, for every delay $\{l \in l_1, \ldots, l_2\}$ (e.g. $l_1 = 0$, $l_2 = L$), the following cost function can be defined:

$$C = \sum_{i=1}^{n} \sum_{j=1,j=1}^{n} \sum_{l=l_1}^{l_2} r_{y_i y_j}^2(l) \quad (13)$$

where the dependence factor for zero-mean signals is:

$$r_{y_i y_j}(l) = E\left\{f\left[y_i(k)\right] g\left[y_j(k+l)\right]\right\} \quad (14)$$

and $f[y]$ and $g[y]$ is a suitably defined function-pair such that: $f[y] = y^3$; $g[y] = y$ (for sub-Gaussian signals) or $f[y] = y$; $g[y] = \tanh(y)$ (for super-Gaussians).

2.3. The ECDA algorithm for the BSD problem.

By a minimization of the goal function (13) a weight update rule (for matrix **W**) can be derived (see section 3), that is a generalization of the CDA rule proposed in [12].

We make a constant diagonal assumption, which prohibits a decay of weights **W** to zero. The diagonal weight elements are kept constant, i.e. equal to

$\mathbf{w}_{ii}^T = [1, 0, \ldots, 0]$ for $i = 1 \ldots n$.

In our ECDA method the weight update rule for every vector $\mathbf{w}_{lm}$ $(l \neq m)$ is as follows:

$$\begin{aligned} \mathbf{w}_{lm}^T = f^{-1}\Big\{-\Big(\sum_{j\neq l}\sum_{c=1}^{n}\sum_{d=1}^{n} \mathbf{A}_{jc}^T \mathbf{R}_{xmxc} \mathbf{R}_{xmxd}^T \mathbf{A}_{jd}\Big)^{-1} \\ \times \Big(\sum_{j\neq l}\sum_{c=1}^{n}\sum_{b\neq m}\sum_{d=1}^{n} \mathbf{A}_{jc}^T \mathbf{R}_{xmxc} \mathbf{R}_{xbxd}^T \mathbf{A}_{jd} f[\mathbf{w}_{lb}]\Big)\Big\} \end{aligned} \quad (15)$$

$\boldsymbol{A}$ is a matrix function depending on $\mathbf{w}$. A particular matrix $\mathbf{A}_{jc}$ is build around such unknown elements $\{w_{jc}(q)\}$ that are used for updating the weight elements with indices l, m and j. $\mathbf{R}$ is a matrix function of input signal cross-dependences. In particular $\mathbf{R}_{xaxc}(l)$ represents a dependence matrix for pairs of signals with given relative time delay of l:

$$r_{x_i x_j}(l) = E\{f[x_i(k)]g[x_j(k+l)]\} \quad (16)$$

The nonlinear functions $f(x)$ and $g(x)$ are defined as explained for equation (2.14). The individual dependences are aggregated to vectors:

$$r_{x_i x_j}^{(l)} = [r_{x_i x_j}(l-q), \cdots, r_{x_i x_j}(l+q)]^T \quad (16)$$

and these vectors are elements of the dependence matrix:

$$\mathbf{R}_{x_a x_c}(l) = \begin{pmatrix} r_{x_a x_c}^{(l)} & \cdots & r_{x_a x_c}^{(l-q)} \\ \vdots & \ddots & \vdots \\ r_{x_a x_c}^{(l+q)} & \cdots & r_{x_a x_c}^{(l)} \end{pmatrix} \quad (18)$$

```
The ECDA algorithm
(1) Init the weights of W = [$\mathbf{w}_i$]
(2) REPEAT
    for $l$=1,...,$n$
        for $m$=1,...,$n$
            {if $l \neq m$ then modify $\mathbf{w}_{lm}$ according to (15)}
    UNTIL the weights are not stable
```

An alternative separation procedure under the constant diagonal mixing assumption is to assure constant power during the separation process (CPA) [12]. In this approach, the decay of weights to zero is not possible due to fixing the independence coefficients for the zero-th delay to some non-zero value K:

$$\boldsymbol{R}_{yiyi}[0] = K, \quad (\text{for } \ i = 1 \ldots n) \quad (19)$$

2.4. Adaptive single-channel blind identification.

The term "blind identification" usually means the identification of a linear system, $\mathbf{y} = \mathbf{x}\ \mathbf{w}^{\mathrm{T}}$, where the delayed input samples are formed to a vector $\mathbf{x} = [x(k), \ldots, x(k\text{-}L)]^{\mathrm{T}}$ and the weights corresponding to delayed signals are $\mathbf{w} = [w_0, \ldots, w_{\mathrm{L}}]^{\mathrm{T}}$. Please note, that the output vector $\mathbf{y}$ from the fist step is now again denoted as input $\mathbf{x}$ to the second step.

The Bussgang algorithm [3,15] minimizes the cost function:

$$J = \frac{E\left\{[f(y(k)) - y(k)]\right\}}{2}, \quad (20)$$

where $f(y)$ should satisfy the Bussgang property over the source signals, i.e. that the cross-correlation between source and its non-linear image has the same shape as the auto-correlation of the source:

$$\frac{E\left\{s(k)\times f(s(k+\Delta))\right\}}{E\left\{s(k)\times f(s(k))\right\}} = \frac{E\left\{s(k)\times s(k+\Delta)\right\}}{E\left\{s(k)\times s(k)\right\}} \tag{21}$$

The update rule for each weight row has the from:

$$\mathbf{w}_j(k+1) = \mathbf{w}_j(k) + \eta\mathbf{x}(k)e_j(k), \tag{22}$$

with

$$e_j(k) = f(y_j(k)) - y_j(k), \quad \mathbf{x}(k) = [x(k), \ldots, x(k-L)]^{\mathrm{T}}$$

The Godard method [3] minimises the cost function:

$$J = \frac{E\{\left[|y(k)|^p - R_p\right]^2 t\}}{2}, \quad \text{where} \quad R_p = \frac{E\left\{|s(k)|^{2p}\right\}}{E\left\{|s(k)|^p\right\}} \tag{23}$$

and p is some positive integer. The update rule is

$$\mathbf{w}_j(k+1) = \mathbf{w}_j(k) + \eta\mathbf{x}(k)e_j^*(k), \tag{24}$$

where

$$e(k) = \frac{\partial J}{\partial y} = |y(k)|^{p-1} sign[y(k)][R_p - |y(k)|^p]. \tag{25}$$

The specific case, when $p = 1$, is a modification of the Sato approach, where the original Sato's cost function was:

$$j = \frac{E\left\{[-y(k) + R_1 sign(y(k))]^2\right\}}{2}. \tag{26}$$

We apply a specific case of the Godard approach, for $p = 2$, which is called the Constant Modulus Algorithm [3]. The cost function:

$$J = \frac{E\{\left[y^2(k) - R_2\right]^2\}}{2} \quad \text{with} \quad R_2 = \frac{E\left\{|s(k)|^4\right\}}{E\left\{|s(k)|^2\right\}} \tag{27}$$

leads to the same form of update rule as (23):

$$\boxed{\begin{aligned} &\mathbf{w}_j(k+1) = \mathbf{w}_j(k) + \eta\mathbf{x}(k)e_j^*(k), \\ \text{but with} \quad &e(k) = y(k)[R_2 - |y(k)|^2]. \qquad (28) \end{aligned}}$$

In practice every output signal from the deconvolution step is divided into time frames, where for every frame the stationary signal assumption can be approximately satisfied.

3. Derivation of the ECDA MBD rule

The diagonal elements of the de-mixing matrix are fixed:

$$\boldsymbol{w}_{ii}^T = [1, 0, \ldots, 0], \quad i = 1, \ldots, n.$$

We assume zero-mean sources (if not, they can always be normalized to zero-mean). The statistical dependence of two outputs is expressed as:

$$r_{yiyj}(l) = E\left\{f[y_i(t)]g[y_j(t-l)]\right\} = 0, \quad \forall\, i \neq j, \ \forall\, l.$$

As specified in (14) we apply the function $f(y) = y^3$ (for sub-Gaussian signals) or $f(y) = y$ (for super-Gaussian signals). Hence: $f[y] = f[w] \times f[x]$.

Similarly, the function $g(y) = y$ or $g(y) = \tanh(y)$. For the second type of the function the decomposition into a product of two functions is only approximately satisfied: $g[y] \cong g[w] \times g[x]$.

Assuming above decomposition, the dependence of outputs y_i (i = 1, ..., n) can be expressed in terms of weights and inputs for given delay value, i.e. i (index of 1-st output in pair), j (index of 2-nd output in given pair), l (the relative delay index):

$$\begin{aligned} r_{yiyj}(l) &\cong \\ &\cong \Bigg[\sum_{a=1}^{n}\sum_{b=0}^{q} f[w_{ia}(b)] \\ &\quad \times f[x_a(k-b)] \sum_{c=1}^{n}\sum_{d=0}^{q} g[w_{jc}(d)] \cdot g[x_c(k+l-d)]\Bigg] \\ &= \sum_{a=1}^{n}\sum_{b=0}^{q}\sum_{c=1}^{n}\sum_{d=0}^{q} f[w_{ia}(b)]g[w_{jc}(d)]r_{xaxc}(l+b-d). \end{aligned} \tag{29}$$

The dependence vectors for the input signals are:

$$\begin{aligned} r_{xixj}(l) &= E\left\lfloor f[x_i(k)]g[x_j(k+l)]\right\rfloor \\ \mathbf{r}_{xixj}(l) &= \left[r_{xixj}(l-q), \ldots, r_{xixj}(l+q)\right]^T. \end{aligned} \tag{30}$$

For two inputs the dependence factors are summarised by the following matrix:

$$\mathbf{R}_{xaxc}(l) = \begin{bmatrix} r_{xaxc}(l) & \cdots & r_{xaxc}(l-q) \\ \vdots & \ddots & \vdots \\ r_{xaxc}(l+q) & \cdots & r_{xaxc}(l) \end{bmatrix}. \tag{31}$$

Let us denote these weights $w_{ij}(q)$ which are used during the calculation of dependence factors for given parameters i,j,l:

$$\boldsymbol{A}_{jc} = \begin{pmatrix} g[w_{jc}(q)] & 0 & \cdots & 0 & 0 \\ g[w_{jc}(q)] & g[w_{jc}(q)] & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ g[w_{jc}(1)] & g[w_{jc}(2)] & \cdots & g[w_{jc}(q)] & 0 \\ g[w_{jc}(0)] & g[w_{jc}(1)] & \cdots & g[w_{jc}(q-1)] & g[w_{jc}(q)] \\ 0 & g[w_{jc}(0)] & \cdots & g[w_{jc}(q-2)] & g[w_{jc}(q-1)] \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & g[w_{jc}(0)] & g[w_{jc}(1)] \\ 0 & 0 & \cdots & 0 & g[w_{jc}(0)] \end{pmatrix}. \tag{32}$$

Thus (29) can be rewritten as:

$$r_{yiyj}(l) = \sum_{a=1}^{n}\sum_{c=1}^{n} f[\mathbf{w}_{ia}^T]\mathbf{A}_{jc}^T\mathbf{r}_{xaxc}(l). \tag{33}$$

The dependence factors for given pair of outputs are aggregated into following vectors (indexed only by i, j):

$$\mathbf{r}_{yiyj}^T = \left[r_{yiyj}(l_1), \ldots, r_{yiyj}(l_2)\right]. \tag{34}$$

The computation of all dependence coefficients can be expressed now by the following vector equation:

$$\mathbf{r}^T_{yiyj} = \sum_{a=1}^{n} \sum_{c=1}^{n} f[\mathbf{w}^T_{ia}] \mathbf{A}^T_{jc} \mathbf{R}_{xaxc}. \tag{35}$$

The criterion for an ideal de-mixing is to achieve zero values of all dependence coefficients: for all output pairs (i, j) and all relative delays (from $-L$ to L). In practice we require to minimize the sum of quadratic formulas over all pairs (i, j):

$$C = \sum_{i \neq j} \mathbf{r}^T_{yiyj} \mathbf{r}_{yiyj}$$

$$C = \Big(\sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} \sum_{a,b,c,d=1}^{n} f[\mathbf{w}^T_{ia}] \mathbf{A}^T_{jc} \mathbf{R}_{x_a x_c} \mathbf{R}^T_{x_b x_d} \mathbf{A}_{jd} f[\mathbf{w}_{ib}] \Big). \tag{36}$$

Hence the update rule is derived from the minimization of C with respect to coefficients $\boldsymbol{w}_{lm}$:

$$\frac{\partial C}{\partial \mathbf{w}_{lm}} = \frac{\partial}{\partial \mathbf{w}_{lm}} \Big(\sum_{j \neq l} \sum_{c=1}^{n} \sum_{d=1}^{n} f[\mathbf{w}^T_{lm}] \mathbf{A}^T_{jc} \mathbf{R}_{xmxc} \mathbf{R}^T_{xmxd} \mathbf{A}_{jd} f[\mathbf{w}_{lm}] \Big) + \frac{\partial}{\partial \mathbf{w}_{lm}} \Big(\sum_{j \neq l} \sum_{c=1}^{n} \sum_{b \neq m} \sum_{d=1}^{n} f[\mathbf{w}^T_{lm}] \mathbf{A}^T_{jc} \mathbf{R}_{xmxc} \mathbf{R}^T_{xbxd} \mathbf{A}_{jd} f[\mathbf{w}_{lb}] \Big) + \frac{\partial}{\partial \mathbf{w}_{lm}} \Big(\sum_{j \neq l} \sum_{a \neq m} \sum_{c=1}^{n} \sum_{d=1}^{n} f[\mathbf{w}^T_{la}] \mathbf{A}^T_{jc} \mathbf{R}_{xaxc} \mathbf{R}^T_{xmxd} \mathbf{A}_{jd} f[\mathbf{w}_{lm}] \Big) \tag{37}$$

$$\frac{\partial C}{\partial \mathbf{w}_{lm}} = 2 \Big(\sum_{j \neq l} \sum_{c=1}^{n} \sum_{d=1}^{n} \mathbf{A}^T_{jc} \mathbf{R}_{xmxc} \mathbf{R}^T_{xmxd} \mathbf{A}_{jd} \Big) f[\mathbf{w}^T_{lm}] \frac{\partial f[\mathbf{w}^T_{lm}]}{\partial \mathbf{w}_{lm}} + 2 \Big(\sum_{j \neq l} \sum_{c=1}^{n} \sum_{b \neq m} \sum_{d=1}^{n} \mathbf{A}^T_{jc} \mathbf{R}_{xmxc} \mathbf{R}^T_{xbxd} \mathbf{A}_{jd} f[\mathbf{w}_{lb}] \Big) \frac{\partial f[\mathbf{w}^T_{lm}]}{\partial \mathbf{w}_{lm}} \tag{38}$$

The minimum of gradient (38) is achieved if:

$$\frac{\partial C}{\partial \mathbf{w}_{lm}} = 0 \tag{39}$$

Equations (38) and (39) are rewritten into:

$$\Big(\sum_{j \neq l} \sum_{c=1}^{n} \sum_{d=1}^{n} \mathbf{A}^T_{jc} \mathbf{R}_{xmxc} \mathbf{R}^T_{xmxd} \mathbf{A}_{jd} \Big) f[\mathbf{w}^T_{lm}] = -\Big(\sum_{j \neq l} \sum_{c=1}^{n} \sum_{b \neq m} \sum_{d=1}^{n} \mathbf{A}^T_{jc} \mathbf{R}_{xmxc} \mathbf{R}^T_{xbxd} \mathbf{A}_{jd} f[\mathbf{w}_{lb}] \Big) \tag{40}$$

and further into:

$$f[\mathbf{w}^T_{lm}] = -\Big(\sum_{j \neq l} \sum_{c=1}^{n} \sum_{d=1}^{n} \mathbf{A}^T_{jc} \mathbf{R}_{xmxc} \mathbf{R}^T_{xmxd} \mathbf{A}_{jd} \Big) \times \Big(\sum_{j \neq l} \sum_{c=1}^{n} \sum_{b \neq m} \sum_{d=1}^{n} \mathbf{A}^T_{jc} \mathbf{R}_{xmxc} \mathbf{R}^T_{xbxd} \mathbf{A}_{jd} f[\mathbf{w}_{lb}] \Big). \tag{41}$$

4. Experiments

The experiments have been performed in the following environment (Fig. 4). A multi-channel sound acquisition card Delta 44 produced by M-Audio [16] was installed in a PCI interface of a PC with Intel Pentium 2.4 GHz processor, with 512 kB cache and FSB bus 533 MHz. Four dynamic microphones C608 produced by Shure have been plugged in (working spectrum 50 - 15 kHz) [17].

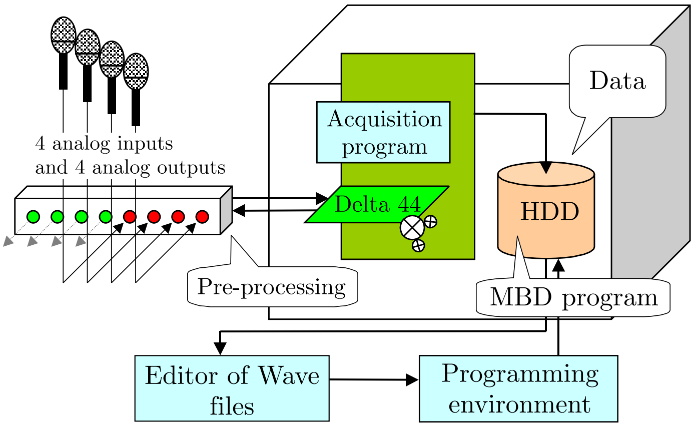

Fig. 4. The test environment

The acquisition software consisted of n-Track Studio (a 24-bit version 3.3) [18] and a file editor Audacity [19]. The sound source was acquired with 16-bit digital quality and with sampling frequency of 44 kHz. Later the signal was re-sampled to a frequency of 12 kHz, in order to lower the computational requirements.

In Figures 5–7 and 8–10 the mixtures and de-mixing results for 3 sources or 4 sources (sound, speech, noise), respectively, are shown. In all tests the first de-mixing step consisted of our ECDA method and the second one – of channel equalisation performed by the Godard method. In all cases the sampling frequency was 12000 Hz and a 16-bit digital sample representation. The number of samples was 30000 – near 2.5 seconds of signal length.

The separation quality was measured in terms of the signal-to-noise ratio, i.e. the relative distance between an estimated source and the original source (after the amplitudes of both the source and output signals were normal-

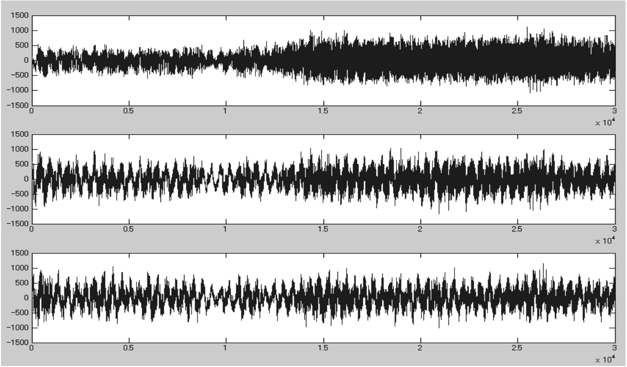

Fig. 5. Three input signals (mixtures of speech sources

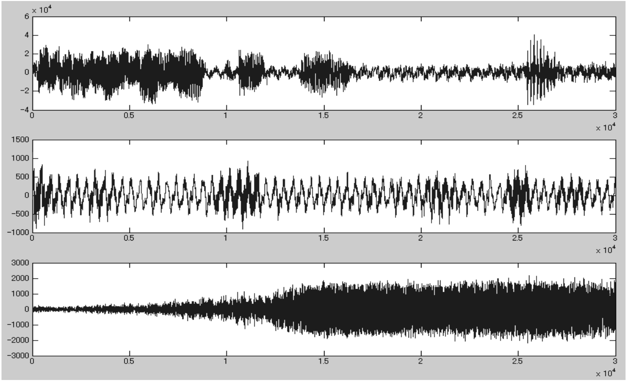

Fig. 6. Three estimated sources after the first de-mixing step (the ECDA method

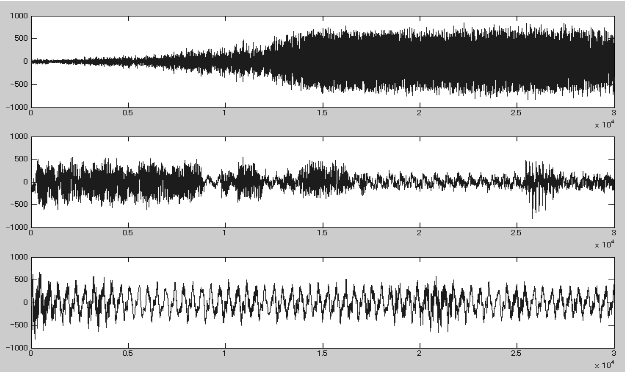

Fig. 7. Three estimated sources after the 2-nd de-mixing (single channel equalization

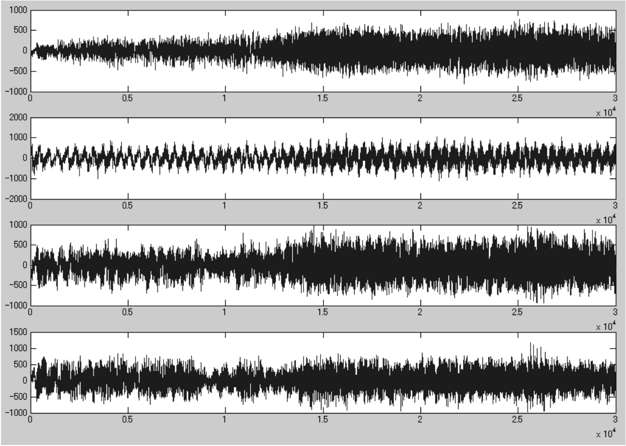

Fig. 8. Four input mixtures for the de-mixing system (natural signals with an additional convolutive mixing)

ized to the interval $\langle -1, 1\rangle$). For the first experiment the SNR was around 19-22 dB for all sources. In the second experiment the SNR was around 15-20 dB. The subjective feeling was quite satisfactory. Obviously, the ordering of sources cannot be determined without additional assumptions and some post-processing. The practical verification will appear in terms of speech recognition – we require the recognition procedure to work comparably well both for the original sources and the reconstructed sources.

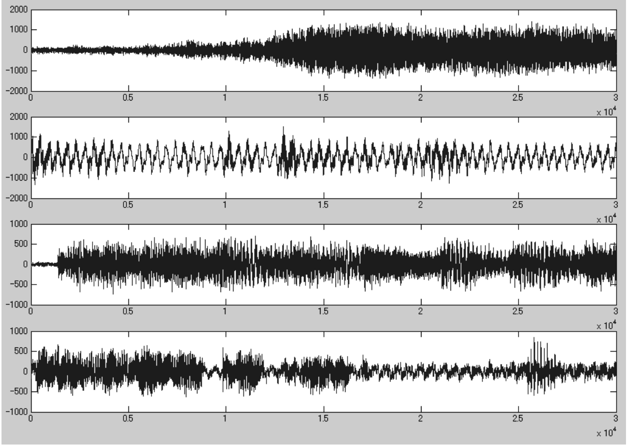

Fig. 9. The 4 estimated sources after the ECDA de-mixing step

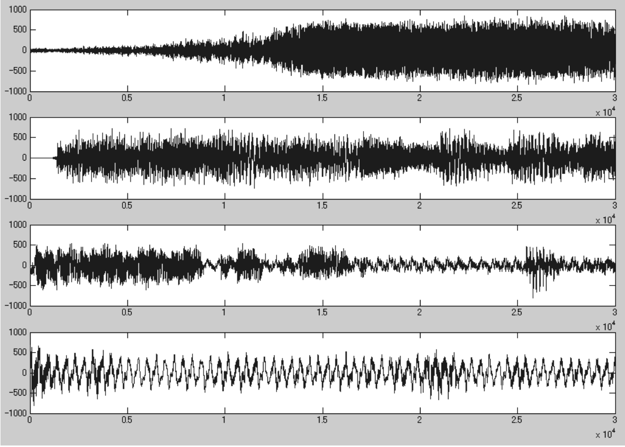

Fig. 10. The 4 estimated sources after the 2-nd de-mixing (single channel equalization)

5. Summary

A two-step approach to many-channel blind deconvolution in time space was proposed and experimentally tested. The results for synthetic convoluted mixtures of real sources very well verify this approach. In case of natural mixtures we face a serious problem of a proper detection of such time windows in the mixtures in which all the sources are contributing to the input mixtures (i.e. they are of non-zero values in these windows). Hence, the proposed approach should be extended to handle the deconvolution of sparse signals.

Acknowledgements. The authors gratefully acknowledge the support by research grant MNiI – 4T11A 003 25.

References

[1] K. Tchoń (ed.), *VIII State Conference of Robotics*, (Polanica Zdrój, Poland, June 2004), WKiŁ, Warszawa, (2005), (in Polish).

[2] C. Zieliński, "A unified formal description of behavioural and deliberative robotic multi-agent systems", *Proc. 7th IFAC International Symposium on Robot Control SYROCO 2003*, Wrocław, Poland, vol. 2, 479-486 (2003).

[3] A. Cichocki and S. Amari, *Adaptive Blind Signal and Image Processing*, John Wiley, Chichester, UK, (2002).

[4] A. Hyvarinen, J. Karhunen and E. Oja, *Independent Component Analysis*, John Wiley & Sons, New York etc., (2001).

[5] N. Murata, S. Ikeda and A. Ziehe, "An approach to blind source separation based on temporal structure of speech signal", *Neurocomputing* 41(4), 1–24 (2001).

[6] W. Kasprzak, "Blind deconvolution of timely correlated sources by gradient descent search", in: *Image Processing and Communications, An International Journal*, edited by ATR Bydgoszcz, 9(1), 33–52 (2003).

[7] P. Smaragdis, "Blind separation of convolved mixtures in frequency domain", *Neurocomputing* 22 (1), 21–34 (1998).

[8] N. Araki et al., "Fundamental limitation of frequency domain blind source separation for convolved mixture of speech", *Proceedings ICASSP2001*, 5, 2737–2740 (2001).

[9] W. Kasprzak and A. Okazaki, "Blind deconvolution of timely-correlated sources by homomorphic filtering in Fourier space", *Fourth Int. Symposium on Independent Component Analysis and Blind Signal Separation - ICA'2003*, Nara, Japan, 2003, NTT Comm. Science Lab., pp. 1029–1034 (2003).

[10] H. Saruwatari, S. Kurita and K. Takeda, "Blind source separation combining frequency-domain ICA and beamforming", *Proceedings ICASSP2001*, 5, 2733–2736 (2001).

[11] T. Nishikawa, H. Saruwatari, K. Shikano, S. Araki and S. Makino, "Multistage ICA for blind source separation of real acoustic convolutive mixture", *Fourth Int. Symposium on Independent Component Analysis and Blind Signal Separation – ICA'2003*, Nara, Japan, NTT Comm. Science Lab., pp. 523–528 (2003).

[12] D.C.B. Chan, *Blind Signal Separation*, Ph.D. thesis, University of Cambridge, Engineering Department, Cambridge, UK, (1997).

[13] W. Kasprzak, A. Cichocki and S. Amari, "Blind source separation with convolutive noise cancellation", *Neural Computing and Applications*, Springer-Verlag London Ltd., vol. 6, 127–141 (1997).

[14] A. Okazaki and W. Kasprzak, "Deconvolution of speech signals from their mixture under constant mixing matrix diagonal", *VIII State Conference of Robotics*, (Polanica Zdrój, czerwiec 2004), WKiL, Warszawa, (2004), (in Polish).

[15] I. Sabala, *Multichannel Deconvolution and Separation of Statistically Independent Signals for Unknown Dynamic Systems*, Ph.D. dissertation, Warsaw University of Technology, Dep. of Electrical Engineering, Warsaw, (1998).

[16] M-Audio Inc., "M-Audio Delta Series 44 User's Manual", M-Audio, http://www.m-audio.com, (2000).

[17] Shure Corporation, "Shure C608 user's manual", Shure Company, (2003).

[18] F. Antonioli, "WWW page of company *FASoft* with program *n-Track Studio*", http://www.fasoft.com

[19] M. Burbeck, J. Haberman and D. Mazzoni, "WWW page of project Audacity", http://audacity.sourceforge.net/.