BLIND SOURCE SEPARATION FOR CONVOLUTIVE MIXTURES BASED ON COMPLEXITY MINIMIZATION

Sébastien de la Kethulle de Ryhove   Ryo Mukai   Hiroshi Sawada   Shoji Makino

NTT Communication Science Laboratories, NTT Corporation
2–4 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619–0237, Japan
{delaketh,ryo,sawada,maki}@cslab.kecl.ntt.co.jp

ABSTRACT

Using algorithmic complexity to perform blind source separation (BSS) was first proposed by Pajunen. This approach presents the advantage of taking the whole signal structure into account to achieve separation, whereas standard ICA-based methods only use either time-correlations or higher order statistics in order to do so. Another advantage of this approach is that no assumptions about the probability distribution of the source signals need to be made. However, although algorithmic complexity based methods have been shown to outperform standard ICA algorithms in the instantaneous BSS case, they haven't been applied to convolutive BSS to the present date. In this paper, we show that it is also possible to use algorithmic complexity as a separating criterion to perform BSS for convolutive mixtures and suggest a method to do so. Testing the proposed method by computer simulation yielded results which are encouraging in terms of SNR performance.

1. INTRODUCTION

Blind source separation (BSS) is a method for recovering a set of source signals from the observation of their mixtures without any prior knowledge about the mixing process. We consider here the convolutive mixture case, i.e. n source signals $\mathbf{s}(t) = (s_1(t), \dots, s_n(t))^T$ are mixed and the corresponding mixtures $\mathbf{x}(t) = (x_1(t), \dots, x_m(t))^T$ observed at m sensors, following $x_j(t) = \sum_{i=1}^{n} \sum_l h_{ji}(l) s_i(t-l)$, where $h_{ji}(l)$ represents the impulse response from source i to sensor j. The goal is to find a separating system consisting of FIR filters $w_{ij}(l)$ of length L to produce separated signals $y_i(t) = \sum_{j=1}^{m} \sum_{l=0}^{L-1} w_{ij}(l) x_j(t-l)$ that are as close as possible to the source signals $s_i(t)$.

Many methods based on independent component analysis (ICA) have been proposed to solve the above BSS problem. We do not review them here, extensive literature on the subject having been published over the past years (see for instance [1, 2, 3]). We however give an outline of the less well known algorithmic complexity based approaches.

The *Kolmogorov (or algorithmic) complexity* $K_U(x^N)$ of a string $x^N = x_1 x_2 \dots x_N$ with respect to a universal computer U is defined as ([4, 5])

$$K_U(x^N) = \min_{p_U(x^N)} |p_U(x^N)|, \tag{1}$$

the minimum length over all programs $p_U(x^N)$ on a universal computer U that print x^N and halt. The universal computer U is henceforth considered fixed and its mention thus omitted (see [4] for more extensive explanations). The BSS problem can then be defined as [5]:

Having observed mixtures $\mathbf{x}(t) = \mathbf{f}(\mathbf{s}(t))$, find a mixing mapping $\hat{\mathbf{f}}$ such that the total complexity of the mapping $\hat{\mathbf{f}}$ and the separated signals $\mathbf{y}(t)$ is minimized.

Applying this definition to convolutive BSS leads to the minimization of the cost function

$$J_K\left[\mathbf{W}(l), \mathbf{y}(t)\right] = K\left[\mathbf{W}^{-1}(l)\right] + \frac{1}{T} K[\mathbf{y}(t)], \tag{2}$$

where the separating system $\mathbf{W}(l) = [w_{ij}(l)]$ is assumed to be invertible and $\mathbf{y}(t) = (y_1(t), \dots, y_n(t))^T$ is the vector of separated signals, its components $y_i(t)$ being of length T for all i.

When using the complexity minimization criterion, no assumptions about the distribution of the source signals need to be made. Moreover, both time correlations and higher order statistics are taken into account to perform BSS, as opposed to standard ICA-based methods which use only one of these two criteria (see [5] for more details).

2. PROPOSED METHOD

We consider here the estimation of only *one* source signal $s_\alpha(t)$, with $\alpha \in \{1, \dots, n\}$. In the remainder of this section, we denote by $s(t)$ the source signal to be estimated (the index α is dropped for more clarity), $y(t)$ is the length T estimate for $s(t)$, and $\mathbf{w}(l) = (w_1(l), \dots, w_m(l))$ denotes the separating FIR filters. We hence have $y(t) = \sum_{j=1}^{m} \sum_{l=0}^{L-1} w_j(l) x_j(t-l)$.

In the case of the above one-unit algorithm, the unmixing system $\mathbf{w}(l)$ is not invertible and the cost function (2) cannot be used due to the presence of the first term, which we choose here to ignore (this is also done in [6] in the linear BSS case). The cost function (2) then becomes

$$J'_K[y(t)] = \frac{1}{T} K[y(t)]. \tag{3}$$

In order to actually perform BSS, we first need to derive an approximative expression $\hat{K}[y(t)]$ for the algorithmic complexity $K[y(t)]$ of a time sequence $y(t)$. This is the object of sec. 2.1 (a similar discussion can also be found in [6]). Finally, we use the cost function

$$J'_{\hat{K}}[y(t)] = \frac{1}{T} \hat{K}[y(t)] \tag{4}$$

to implement BSS. Sec. 2.2 is devoted to the calculation of the gradient of (4) using the approximation $\hat{K}[y(t)]$ for $K[y(t)]$ obtained in sec. 2.1. In sec. 2.3, we discuss how to impose a normalization contraint on $\mathbf{w}(l)$ in order to avoid solutions of the form $\mathbf{w}(l) = 0$.

2.1. Algorithmic Complexity of a Time Sequence

Consider a time sequence $y(t) = (y(0), \dots, y(T-1))$. It can be shown (see [4]) that if its samples $y(0), \dots, y(T-1)$ are i.i.d. according to the probability density function $p(y)$ of a random variable Y, then

$$\lim_{T \to \infty} \frac{1}{T} K[y(t)] = H[y(t)], \tag{5}$$

where the notation $H[y(t)]$ is used for the entropy $H(Y)$ of the random variable Y (for convenience, a similar notation will be used in all expressions involving expectations in the remainder of this paper).

However, if $y(t)$ is a natural signal, there will in general be dependencies among its samples, and the above theorem cannot be used. Therefore, in order to help remove these dependencies, we use linear prediction (see [7]) to find an estimate $\hat{y}(t)$ for $y(t)$ following

$$\hat{y}(t) = \sum_{\tau=1}^{\tau_0} \alpha_\tau y(t-\tau) \tag{6}$$

and define the sequence of the residuals $\delta y(t)$ of $y(t)$ as

$$\delta y(t) \triangleq y(t) - \hat{y}(t), \qquad t = \tau_0, \ldots, T-1, \tag{7}$$

the idea being to approximate the complexity of the sequence $y(t)$ by that of the sequence $\delta y(t)$. Note that in order to perform linear prediction, the process $y(t)$ must be wide-sense stationary. Moreover, if $y(t)$ can be modeled as an AR process of order τ_0, then the sequence $\delta y(t)$ is white and equal to the innovations process of $y(t)$. If its samples are also i.i.d. according to the probability density function $p(\delta y)$ of a random variable δY, then the equality

$$\lim_{T\to\infty} \frac{1}{T-\tau_0} K[\delta y(t)] = H[\delta y(t)] \tag{8}$$

holds, where $H[\delta y(t)]$ denotes the entropy of the random variable δY. However, it is difficult to find natural sequences $y(t)$ that can exactly be modeled as AR processes, and even in that case, there is no guarantee that the samples of the innovations process $\delta y(t)$ will be mutually independent. Equality (8) is thus only approximative. Nevertheless, for natural signals $y(t)$, it is reasonable to assume that (8) is closer to being verified than (5) since linear prediction, as used in (6) and (7), helps remove dependencies. The complexity of the sequence $y(t)$ is thus approximated by

$$K[y(t)] \approx \frac{T}{T-\tau_0} K[\delta y(t)] \approx TH[\delta y(t)], \tag{9}$$

where the second approximation was obtained using (8). Note that the complexity of the prediction coefficients $(\alpha_1, \ldots, \alpha_{\tau_0})$ has been neglected in (9).

Let us now consider the problem of the estimation of the entropy of the residual $H[\delta y(t)]$ which appears in (8) and (9). It is useful to normalize the residual δy to unit variance in order to remove the effects of scaling. Denoting by σ_δ the variance of the residual, we have

$$H[\delta y(t)] = H\left[\frac{\delta y(t)}{\sigma_\delta}\right] + \log \sigma_\delta \tag{10}$$

If a good approximation (denoted $G(\cdot)$) of the negative log density of the pdf of the residual normalized to unit variance $-\log\left[p(\frac{\delta y}{\sigma_\delta})\right]$ is known, we can easily approximate $H[\delta y(t)]$ as

$$H[\delta y(t)] \approx E\left[G\left(\frac{\delta y}{\sigma_\delta}\right)\right] + \log \sigma_\delta. \tag{11}$$

In most cases, $p(\frac{\delta y}{\sigma_\delta})$ is unknown and we set $G(\cdot)$ equal to the negative log density of a generic super- or subgaussian random variable (depending on the nature of the residual). We finally obtain the following approximation for the per sample complexity $K_T[y(t)] \triangleq \frac{1}{T}K[y(t)]$ of the time sequence $y(t)$:

$$\hat{K}_T[y(t)] = E\left[G\left(\frac{\delta y}{\sigma_\delta}\right)\right] + \log \sigma_\delta. \tag{12}$$

2.2. Gradient Descent

Now that an approximation for the algorithmic complexity of a sequence $y(t)$ has been derived, let us return to the problem of blind source separation of convolutive mixtures, where we extract *one* component $y(t) = \sum_{j=1}^{m} \sum_{l=0}^{L-1} w_j(l) x_j(t-l) = \sum_{l=0}^{L-1} \mathbf{w}^T(l)\mathbf{x}(t-l)$, with $\mathbf{w}(l) = (w_1(l), \ldots, w_m(l))^T$ and $\mathbf{x}(t) = (x_1(t), \ldots, x_m(t))^T$. We now have to find the FIR filters $\mathbf{w}(l)$ of length L such that the cost function given in (4),

$$J'_{\hat{K}}[y(t)] = \hat{K}_T[y(t)], \tag{13}$$

where $\hat{K}_T[y(t)]$ is evaluated using (12), is minimized. We thus calculate the derivative $\frac{\partial}{\partial \mathbf{w}(q)} J'_{\hat{K}}[y(t)]$ in order to perform gradient descent on $J'_{\hat{K}}[y(t)]$. After some basic algebraic manipulations, we obtain:

$$\begin{aligned}\frac{\partial J'_{\hat{K}}[y(t)]}{\partial \mathbf{w}(q)} &= \frac{\partial}{\partial \mathbf{w}(q)} \hat{K}_T\left[\sum_{l=0}^{L-1} \mathbf{w}^T(l)\cdot \mathbf{x}(t-l)\right] = \\ & E\left[\delta \mathbf{x}(t-q)\ g\left(\frac{\delta y\ (t)}{\sigma_\delta}\right)\right] \\ &+ \frac{1}{\sigma_\delta}\left\{1 - \frac{1}{\sigma_\delta} E\left[\delta y\ (t)\, g\left(\frac{\delta y\ (t)}{\sigma_\delta}\right)\right]\right\} \frac{\partial}{\partial \mathbf{w}(q)}\ \sigma_\delta \\ &- \frac{1}{\sigma_\delta}\sum_{\tau=1}^{\tau_0}\left\{ E\left[y(t-\tau)\, g\left(\frac{\delta y\ (t)}{\sigma_\delta}\right)\right] \cdot \frac{\partial}{\partial \mathbf{w}(q)}\ \alpha_\tau\right\},\end{aligned} \tag{14}$$

with $\delta\mathbf{x}(t) \triangleq \mathbf{x}(t) - \sum_{\tau=1}^{\tau_0} \alpha_\tau \mathbf{x}(t-\tau)$, $\delta y(t) = \sum_{l=0}^{L-1} \mathbf{w}^T(l) \left[\mathbf{x}(t-l) - \sum_{\tau=1}^{\tau_0} \alpha_\tau \mathbf{x}(t-l-\tau)\right]$, σ_δ the variance of $\delta y(t)$, $\alpha_1 \ldots \alpha_{\tau_0}$ the linear prediction coefficients obtained using $y(t)$ and $g(\cdot)$ the derivative of $G(\cdot)$. To actually use this expression in a gradient descent algorithm, we need to evaluate $\frac{\partial}{\partial \mathbf{w}(q)}\ \sigma_\delta$ and $\frac{\partial}{\partial \mathbf{w}(q)} \alpha_\tau$ for all $\tau \in \{1, \cdots, \tau_0\}$. The expressions for $\alpha_1 \ldots \alpha_{\tau_0}$ and σ_δ being quite complex (especially for large τ_0), this is not an easy task.

We hence restrict ourselves to the case $\tau_0 = 1$ (first order linear prediction). In this case, the expressions for $\frac{\partial}{\partial \mathbf{w}(q)}\ \sigma_\delta$ and $\frac{\partial}{\partial \mathbf{w}(q)} \alpha_1$ (which are given in the appendix) are reasonably simple.

2.3. Normalization Constraint

To avoid solutions of the kind $\mathbf{w}(l) = 0$, we need to impose a normalization constraint on $\mathbf{w}(l)$. We propose two different possibilities. A first option consists in adding a term of the form $-\mu \log \sigma_y$ to $J'_{\hat{K}}[y(t)] = \hat{K}_T[y(t)]$, which yields the cost function

$$J'_{\hat{K},\mathrm{CL}}[y(t)] = \hat{K}_T[y(t)] - \mu \log \sigma_y. \tag{15}$$

For $\mu > 0$, this increases the cost of solutions with small values of σ_y^2 (which should be avoided). Note also that the value of the cost function is left almost unchanged if $\sigma_y^2 \approx 1$. Adapting the value of μ so as to have $\sigma_y^2 \approx 1$ using a simple control loop proved to work well in practice: a minimum of the cost function satisfying $\sigma_y^2 \approx 1$ was always attained (see sec. 4).

Computer simulations also showed that, at least in the case of speech mixtures, performing constraint free gradient descent on $J'_{\hat{K}}[y(t)] = \hat{K}_T[y(t)]$ actually seemed to work as well: although both σ_y^2 and the amplitude of $\mathbf{w}(l)$ decrease during minimization, the latter is completed before either σ_y^2 or $\mathbf{w}(l)$ become unacceptably small. This is our second method.

In the sequel, we respectively refer to our two methods of dealing with the normalization problem as the “control loop” and “unconstrained” methods.

3. CONVOLUTIVE SPEECH MIXTURES

In this section, we show how to apply the method outlined in sec. 2 to the particular case of BSS for convolutive speech mixtures. We thus assume that sequences $y(t)$ and $\mathbf{x}(t)$ represent speech signals sampled at a given frequency f_s.

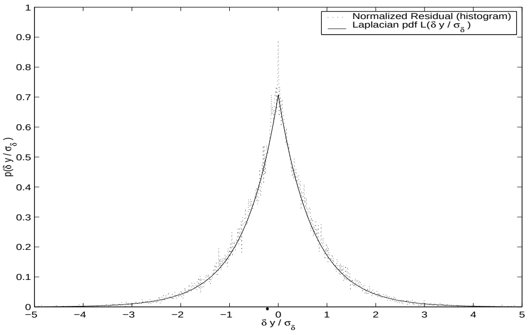

Fig. 1. Probability density function of the residual (normalized to unit variance) $p(\delta y/\sigma_\delta)$ as estimated using histograms. The Laplacian pdf $L(\delta y/\sigma_\delta) = \frac{\sqrt{2}}{2} \cdot e^{-\sqrt{2}\,|\delta y/\sigma_\delta|}$ is a good approximation for $p(\delta y/\sigma_\delta)$.

3.1. Stationarity and AR model

The properties of speech have already been extensively analyzed: it has among others been shown that it can be considered stationary during time intervals of $T_{\text{stat}} \approx 30$ ms. [8] and that it can accurately be modeled as an AR process [9], the order of which depends on the particular application in view.

In order to compute the approximation of the per sample complexity $\hat{K}_T[y(t)]$ of $y(t)$, we need to use linear prediction, with the underlying assumption that $y(t)$ is wide-sense stationary. Therefore, since $T_{\text{stat}} \approx 30$ ms., we split $y(t)$ into M segments of length $L_{\text{stat}} = f_s \cdot T_{\text{stat}}$ following

$$y(t) = \sum_{i=0}^{M-1} y^{(i)}(t), \tag{16}$$

where

$$y^{(i)}(t) = \begin{cases} y(t) & \text{if } i\,L_{\text{stat}} \leq t < (i+1)\,L_{\text{stat}} \text{ and } t < T \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

If the M segments $y^{(i)}(t)$ are mutually independent, it is reasonable to assume that $K[y(t)] \approx \sum_{i=0}^{M-1} K[y^{(i)}(t)]$ (see [5]). The gradient of $\hat{K}_T[y(t)]$ can then be approximately evaluated as

$$\frac{\partial}{\partial \mathbf{w}(q)} \hat{K}_T[y(t)] \approx \sum_{i=0}^{M-1} \frac{\partial}{\partial \mathbf{w}(q)} \hat{K}_T\left[y^{(i)}(t)\right]. \tag{18}$$

After splitting $\mathbf{x}(t)$ into M segments $\mathbf{x}^{(i)}(t)$, proceeding as with $y(t)$ in (16) and (17), equation (14) – where $\mathbf{x}(t)$ is replaced by $\mathbf{x}^{(i)}(t)$ – can be used to evaluate each one of the terms in the above sum. Linear prediction is also performed separately for each of the M segments $y^{(i)}(t)$ yielding M sequences $\delta y^{(i)}(t)$ which are used instead of $\delta y(t)$ in (14). Note however that the equality

$$y^{(i)}(t) = \sum_{l=0}^{L-1} \mathbf{w}^T(l)\mathbf{x}^{(i)}(t-l) \tag{19}$$

is not quite exact for values of t close to $i \cdot L_{\text{stat}}$ or $(i+1) \cdot L_{\text{stat}}$.

Since speech can be modeled as an AR process, we can reasonably assume that the sequences $\delta y^{(i)}(t)$ are close to being white, and hence that their samples are close to being uncorrelated. This is a necessary condition for having i.i.d. sequences $\delta y^{(i)}(t)$. We can thus hope that (8), where $\delta y(t)$ is replaced by $\delta y^{(i)}(t)$, is approximately verified for each sequence $\delta y^{(i)}(t)$.

3.2. Approximation of the pdf of the Residual

Table 1. Experimental conditions

Source signal length	6 s.
Direction of sources	120° and 50° (two sources)
Inter-sensor spacing	$d = 4\,\text{cm}$. (two sensors)
Reverberation time	$T_r = 130\,\text{ms}$.
Sampling frequency	$f_s = 8\,\text{kHz}$.

In this section, we consider the problem of the evaluation of the negative log density of the pdf of the residual normalized to unit variance $-\log\left[p(\frac{\delta y}{\sigma_\delta})\right]$. Evaluating the probability density $p(\frac{\delta y}{\sigma_\delta})$ using histograms (see Fig. 1) shows that, at least in the case of the parameters specified in sec. 4, it is reasonable to assume that $-\log\left[p(\frac{\delta y}{\sigma_\delta})\right] \approx \frac{1}{2}\log 2 + \sqrt{2}\left|\frac{\delta y}{\sigma_\delta}\right|$, i.e. the negative log-density of a Laplacian random variable with $\lambda = \sqrt{2}$. [1] This is not a surprising result since it is a well known fact that speech is Laplacian distributed. Note however that $-\log\left[p(\frac{\delta y}{\sigma_\delta})\right]$ is function of many different parameters (mixing system, separating system, order of the linear predictor, source signals, iteration number, ...) and in some cases may not be well approximated by $\frac{1}{2}\log 2 + \sqrt{2}\left|\frac{\delta y}{\sigma_\delta}\right|$.

4. COMPUTER SIMULATIONS

The method outlined in secs. 2 and 3 was tested by computer simulation. One source signal was estimated from two two-signal mixtures, this having been done for twelve different source signal combinations. The experimental procedure was the following: the mixing system's impulse responses $\{h_{ij}(l),\ i,j = 1\ldots 2\}$ were first measured in a real room using the experimental conditions summarized in Table 1. Twelve different combinations $(s_1(t), s_2(t))$ of speech signals produced by two male and two female speakers were then mixed following $x_j(t) = \sum_{i=1}^{2}\sum_l h_{ji}(l)s_i(t-l)$, and the proposed one-unit algorithm subsequently applied to extract one speech signal from each of the twelve mixtures. We used a first order linear predictor ($\tau_0 = 1$), a stationarity time $T_{\text{stat}} = 125\,\text{ms}$. and separating filters $\mathbf{w}(l) = (w_1(l), w_2(l))$ of length $L = 256$ taps. Although speech signals can only be considered stationary for time intervals of approximately $30\,\text{ms}$., we set $T_{\text{stat}} = 125\,\text{ms}$. (corresponding to $L_{\text{stat}} = 1000$ taps at $f_s = 8\,\text{kHz}$) in order to be able to learn filters $\mathbf{w}(l)$ of acceptable length (this being because we ideally must have $L \ll L_{\text{stat}}$). Using $T_{\text{stat}} = 125\,\text{ms}$. proved to be acceptable for our complexity evaluation purposes. Moreover, $\mathbf{w}(l)$ was initially set to $\mathbf{w}_{\text{init}}(l) = \left(\delta(\frac{L}{2}), -\delta(\frac{L}{2})\right)$, we used a step size of 1.0×10^{-3}, and two different methods ("control loop" and "unconstrained") were used to deal with the normalization problem (see sec. 2.3).

The results are displayed in Table 2. The SDR (between the original signal $s(t)$ and the extracted signal $y(t)$) was measured using the method proposed in [10]. Both the "control loop" and "unconstrained" versions of the suggested algorithm achieved a good SNR improvement, but the extracted signal was often severely distorted. In the "control loop" version, the distortion was due to filtering (both high-pass filtering and low-pass filtering were observed), whereas in the "unconstrained" version it was due to whitening (see Fig. 2). We observed that the distortion introduced by high- or low-pass filtering was much more unpleasant than the one introduced by whitening when it came to listening. Therefore, we believe that the "unconstrained" version is more suitable than the "control loop" version for convolutive speech mixture BSS. Table 2 also shows that the algorithm performance depends on the nature of the input signals: for instance, in the "control loop" ver-

[1] The generic probability density function of a zero mean Laplacian random variable X is given by $p(x) = \frac{\lambda}{2}e^{-\lambda|x|}$.

	CONTROL LOOP			UNCONSTRAINED		
Speaker Combin.	Ext. Sig.	SNR imp. (dB)	Final SDR (dB)	Ext. Sig.	SNR imp. (dB)	Final SDR (dB)
M1-M2	M1	31.73	-3.29	M1	23.91	-5.59
M2-M1	M2	11.60	-16.88	M1	22.62	-4.18
M1-F1	M1	25.59	-3.03	M1	17.82	-4.23
F1-M1	M1	31.60	-5.55	F1	8.44	-15.36
M1-F2	M1	13.58	-2.12	M1	3.78	-4.16
F2-M1	M1	20.60	-6.05	F2	16.09	-15.15
M2-F1	M2	10.96	-16.41	F1	20.09	-8.18
F1-M2	M2	13.22	-17.19	F1	20.86	-7.57
M2-F2	M2	4.71	-15.11	F2	26.43	-11.63
F2-M2	M2	5.77	-16.36	F2	21.82	-10.61
F1-F2	F1	17.61	-18.21	F1	1.79	-2.51
F2-F1	F1	24.13	-12.92	F2	8.89	-17.98
Average		17.59	-11.09		16.05	-8.92

Table 2. SNR improvement and final SDR obtained with the proposed algorithm ("control loop" and "unconstrained" versions) for twelve different speech signal combinations. $T_r = 130\,\mathrm{ms}$.

sion, the results were much better for combination M1-M2 than for combination F1-F2 .

The fact that the SDR be negative might seem strange at first sight. Note however that we are trying to recover the original signal $s(t)$, which is substantially more difficult than recovering a reverberated version of $s(t)$ (observed at one of the sensors), as in most traditional convolutive BSS algorithms based on ICA. Bearing this in mind, an SDR of -4 dB to -5 dB – comparable to the distortion due to the reverberation time $T_r = 130\,\mathrm{ms}$. – is acceptable. Nonetheless, when the SDR is below -10 dB, the extracted signal, especially in the "control loop" version of the algorithm, can become very unpleasant to hear. Still, we believe that algorithmic complexity is a concept powerful enough to allow to recover the original signal $s(t)$ and will attempt, in future work, to improve the SDR performance by using approximations of the cost function (2) which include the term $K[\mathbf{W}^{-1}(t)]$ (neglected in this paper).

5. CONCLUSIONS

We proposed a method to use algorithmic complexity as a separating criterion to perform BSS for convolutive mixtures. The main advantage of this method over standard ICA algorithms is that, to achieve separation, the whole signal structure is taken into account instead of only time correlations or higher order statistics. The experimental results obtained thus far are encouraging in terms of SNR performance although the extracted signal is often severely distorted. We believe that this problem can be solved by using approximations of the cost function (2) which include the term $K[\mathbf{W}^{-1}(t)]$. This will be the subject of future research.

A. APPENDIX

If we define $\gamma(p) \triangleq E[\,y(t+p)\;\;y(t)\,]$ and $\mathbf{A}_p \triangleq E[\,\mathbf{x}(t+p)\;\mathbf{x}^T(t)\,]$, we have $\sigma_\delta^2 = \frac{\gamma^2(0)-\gamma^2(1)}{\gamma(0)}$ and $\alpha_1 = \frac{\gamma(1)}{\gamma(0)}$.

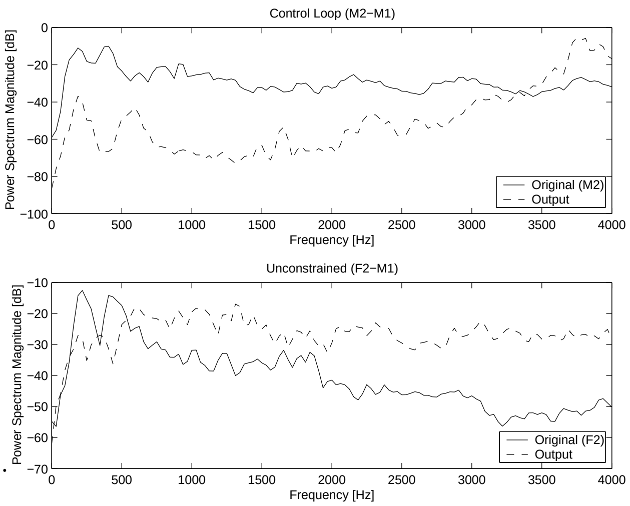

Fig. 2. Power Spectrum Magnitude of the original and extracted signals for the "control loop" (Speaker Combination: M2-M1, top) and "unconstrained" (Speaker Combination: F2-M1, bottom) versions of the proposed algorithm. $T_r = 130\,\mathrm{ms}$.

The derivatives $\frac{\partial}{\partial\mathbf{w}(q)}\,\sigma_\delta$ and $\frac{\partial}{\partial\mathbf{w}(q)}\,\alpha_1$ respectively read

$$\frac{\partial}{\partial\mathbf{w}(q)}\,\sigma_\delta = \frac{1}{\sigma_\delta}\left\{\sum_{l=0}^{L-1}\mathbf{A}_{l-p}\mathbf{w}_l - \frac{\gamma(1)}{\gamma^2(0)}\cdot\right.$$
$$\left.\left[\gamma(0)\sum_{l=0}^{L-1}(\mathbf{A}_{l-p+1}+\mathbf{A}_{l-p-1})\,\mathbf{w}_l\;-\;\gamma(1)\sum_{l=0}^{L-1}\mathbf{A}_{l-p}\mathbf{w}_l\right]\right\} \quad (20)$$

and

$$\frac{\partial}{\partial\mathbf{w}(q)}\,\alpha_1 = \frac{1}{\gamma^2(0)}\left\{\gamma(0)\sum_{l=0}^{L-1}(\mathbf{A}_{l-p+1}+\mathbf{A}_{l-p-1})\,\mathbf{w}_l\right.$$
$$\left.-\;2\,\gamma\,(1)\;\sum_{l=0}^{L-1}\mathbf{A}_{l-p}\mathbf{w}_l\right\}. \quad (21)$$

B. REFERENCES

[1] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, John Wiley & Sons, 2001.

[2] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, no. 6, pp. 1129–1159, 1995.

[3] K. Matsuoka, M. Ohya, and M. Kawamoto, "A neural net for blind separation of nonstationary signals," *Neural Networks*, vol. 8, no. 3, pp. 411–419, 1995.

[4] T. Cover and J. Thomas, *Elements of Information Theory*, John Wiley & Sons, 1991.

[5] P. Pajunen, "Blind source separation using algorithmic information theory," in *Proc. of Independence and Artificial Neural Networks Workshop (I&ANN'98)*, Feb. 1998, pp. 26–31.

[6] A. Hyvärinen, "Complexity pursuit: Separating interesting components from time-series," *Neural Computation*, vol. 13, no. 4, pp. 883–898, 2001.

[7] J. Proakis and D. Manolakis, *Digital Signal Processing*, Prentice Hall, 1996.

[8] A. Oppenheim and R. Schafer, *Discrete-time Signal Processing*, Prentice Hall, 1998.

[9] D. Manolakis, V. Ingle, and S. Kogon, *Statistical and Adaptive Signal Processing*, McGraw-Hill, 2000.

[10] R. Gribonval, E. Vincent, C. Févotte, and L. Benaroya, "Proposals for performance measurement in source separation," in *Proc. ICA 2003*, April 2003, pp. 763–768.