Hindawi Publishing Corporation
ISRN Signal Processing
Volume 2013, Article ID 218651, 17 pages
http://dx.doi.org/10.1155/2013/218651

Research Article

Extraction of Correlated Sparse Sources from Signal Mixtures

M. S. Woolfson,[1] C. Bigan,[2] J. A. Crowe,[1] and B. R. Hayes-Gill[1]

[1] Electrical Systems and Optics Division, Faculty of Engineering, The University of Nottingham, Nottingham NG7 2RD, UK
[2] Engineering Science Department, Ecological University of Bucharest, Bd. Vasile Milea nr. 1G, Sector 6, 061341 Bucharest, Romania

Correspondence should be addressed to M. S. Woolfson; malcolm.woolfson@nottingham.ac.uk

Received 6 November 2012; Accepted 3 December 2012

Academic Editors: S. Callegari, W.-L. Hwang, S. Lee, and P. Ramaswamy

A blind source separation method is described to extract sources from data mixtures where the underlying sources are sparse and correlated. The approach used is to detect and analyze segments of time where one source exists on its own. The method does not assume independence of sources and probability density functions are not assumed for any of the sources. A comparison is made between the proposed method and the Fast-ICA and Clusterwise PCA methods. It is shown that the proposed method works best for cases where the underlying sources are strongly correlated because Fast-ICA assumes zero correlation between sources and Clusterwise PCA can be sensitive to overlap between sources. However, for cases of sources that are sparse and weakly correlated with each other, there is a tendency for Fast-ICA and Clusterwise PCA to have better performances than the proposed method, the reason being that these methods appear to be more robust to changes in input parameters to the algorithms. In addition, because of the deflationary nature of the proposed method, there is a tendency for estimates to be more affected by noise than Fast-ICA when the number of sources increases. The paper concludes with a discussion concerning potential applications for the proposed method.

1. Introduction

One general problem in signal processing is the extraction of individual source signals $\{s_j[n]\}$ from measurements $\{z_i[n]\}$ that are linear combinations of these sources:

$$z_i[n] = \sum_{j=1}^{N} A_{ij} s_j[n] \quad (i = 1, 2, \ldots, S), \tag{1}$$

where $\{A_{ij}\}$ are the mixing coefficients, S is the number of sets of measurement data, and there are N underlying sources. In the case where both the sources and mixing coefficients are unknown, this problem comes under the heading of blind source separation (BSS). There are many applications in this area [1–8].

There are various approaches to extracting the underlying sources $\{s_i\}$. For example, in Independent Component Analysis (ICA), the aim is to determine a transformation of the data that maximizes the negentropy [9]. In the approach designed in [10], the aim is to minimize the cross-cumulants.

Now in some applications, the underlying sources can be approximated as sparse; that is, each source has nonzero values for a finite segment of time, with the other sources having zero values. In practice, this definition can be approximated by each source being dominant over other sources for at least one segment of time. Various specialized BSS methods have been derived for the case of sparse sources [11–18]. In one approach [12], the sparsity of a signal $\{x[k]\}$ is defined over K data points by

$$S(x) = \frac{\sqrt{K}\sqrt{\sum_{k=1}^{K}(x[k])^2}}{\sum_{k=1}^{K}|x[k]|}. \tag{2}$$

Assuming two sources, the aim is to find a transformation on the data $\{z_1[k], z_2[k]\}$ such that the sparsity is maximized. The extension to three or more sources is also discussed. Another approach is based on the fact that regions of the phase plot of the data $\{z_i[k]\}$, where one source exists on its own, are represented by points that form a straight line. These directions are found using clustering techniques on the phase plot. In [11], Minor Component Analysis (MCA) is applied to the clusters to find elements of the separating matrix $\mathbf{B}$ so that the sources can be directly estimated from the data from

$$\mathbf{s} = \mathbf{B}\mathbf{z}. \tag{3}$$

This method will be referred to as the Clusterwise PCA method.

In [17] a method to separate out sparse sources from signal mixtures was presented where the mixing coefficients are unknown as well as the sources. This standard Blind Source Separation (BSS) problem was addressed by analysis of localized segments of the phase space trajectory where one source dominates. The method was found to have a comparable performance to the standard Fast-Independent Component Analysis (Fast-ICA) [9] method for mixtures of sparse sources. Another assumption made by the proposed method was that the underlying sources are uncorrelated, an assumption that is also made by the Fast-ICA method [9] and some other, although not all, BSS methods. However, as the method proposed in [17] analyzes local features of phase space, it makes no assumption about the probability distributions of the underlying sources unlike some BSS methods which make such assumptions; hence if the underlying sources had Gaussian distributions this would be no problem for the method in [17], whilst an approach such as Fast-ICA may encounter difficulty in this situation. In this paper, the method presented in [17] is extended to the case where the underlying sources are sparse *and* correlated. It will be shown that the "prewhitening" stage that is used by many standard BSS and ICA methods is not required here and that the proposed method can work better than the Fast-ICA and Clusterwise PCA methods, for cases where the underlying sources are sparse and strongly correlated. The proposed method can be more sensitive to noise than two other methods tested and can also be more sensitive to changes in input parameters. In particular, for sources that are weakly correlated or uncorrelated, the proposed method offers no advantages compared with Clusterwise PCA and Fast-ICA.

2. Theory

2.1. Minimum Heading Change (MHC) Method. To motivate the approach used in this paper, an example will be shown where the two sources before mixing are shown in Figure 1.

It can be seen that source A is zero for sample numbers larger than 19, whilst source B is zero for sample numbers less than 8. Hence, these two sources can be considered as sparse. Now suppose that the two sources are mixed linearly so that the mixed data looks like Figure 2.

It can be seen that because of the sparsity of the two underlying sources, there are segments of each mixture which depend on one source only; these segments are indicated in Figure 2.

One can identify segments of data where s_1 exists on its own by plotting data input 2, z_2 against data input 1, z_1. In Figure 3 this is done for the example in Figure 2; in the following discussion, we associate source B in Figure 1 with s_1 and source A with s_2.

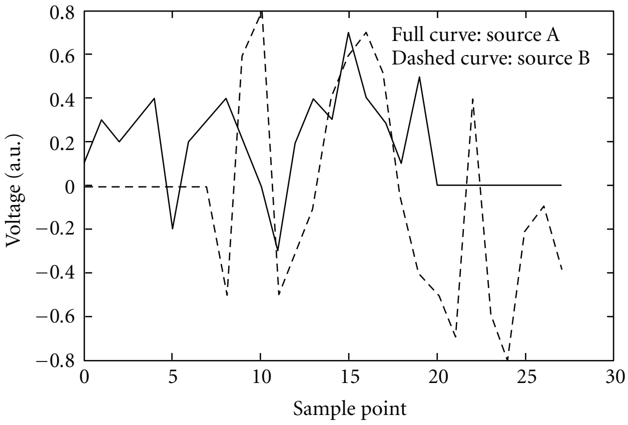


Figure 1: Two source signals.

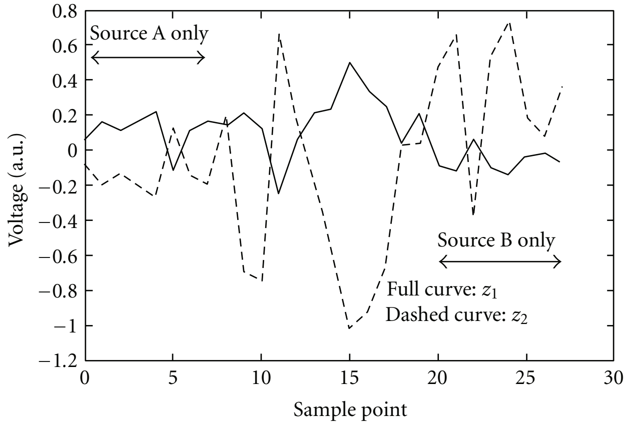


Figure 2: Mixed data.

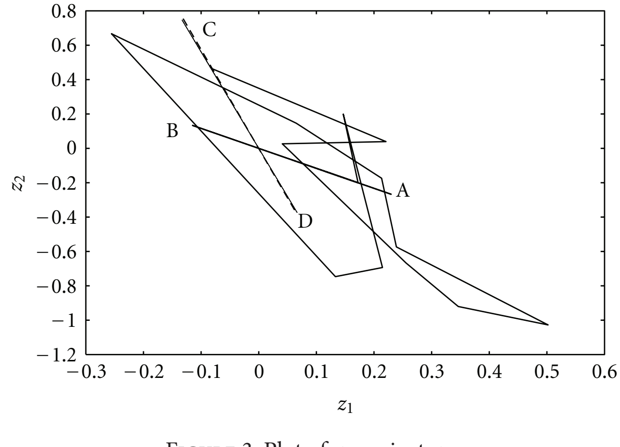


Figure 3: Plot of z_2 against z_1.

In [17], the estimation of sparse sources from mixtures, looking for straight line segments in the phase plots, was described. A summary of the method is described here.

Suppose that there are two mixtures of two sources. Then, from (1),

$$z_1 = A_{11}s_1 + A_{12}s_2,$$
$$z_2 = A_{21}s_1 + A_{22}s_2, \tag{4}$$

where the explicit dependences of the measurements and the sources on sample number have been dropped for notational convenience. Source s_1 is associated with the direction (A_{11}, A_{21}) in the phase plot with source s_2 being associated with the direction (A_{12}, A_{22}). Now for the specific case of mixtures of sources that are sparse, these directions are visible in the phase plot; for example, AB in Figure 3 corresponds to source A on its own in Figure 2, whilst CD (shown dashed in Figure 3) corresponds to source B in Figure 2; this information can be used to estimate the underlying sources.

Suppose that source s_1 is sparse and that sources s_1 and s_2 can be associated with unit vectors in the phase plot $\widehat{\mathbf{R}}_1$ and $\widehat{\mathbf{R}}_2$, respectively. One can think of the plot of z_2 against z_1 as analogous to a radar plot and vectors $\widehat{\mathbf{R}}_1$ and $\widehat{\mathbf{R}}_2$ as "heading" vectors. Any point in the phase plot can be associated with a vector:

$$\mathbf{z} = s_1\mathbf{R}_1 + s_2\mathbf{R}_2 = \sum_{i=1}^{2} s_i R_i \widehat{\mathbf{R}}_i, \tag{5}$$

where the vector magnitude $R_i\,(i = 1, 2)$ depends on the mixing coefficients and $\mathbf{R}_i = R_i\widehat{\mathbf{R}}_i$. Note that $\widehat{\mathbf{R}}_1$ and $\widehat{\mathbf{R}}_2$ are not necessarily orthogonal.

Now suppose that three consecutive points in the phase plot are detected that are collinear corresponding to $\widehat{\mathbf{R}}_1$. One can make an estimate of the component $s_1, \widetilde{s}_1$, by resolving the vector $\mathbf{z}$ in the direction of $\widehat{\mathbf{R}}_1$:

$$\widetilde{s}_1 = \mathbf{z} \cdot \widehat{\mathbf{R}}_1 = \sum_{i=1}^{2} s_i R_i \widehat{\mathbf{R}}_i \cdot \widehat{\mathbf{R}}_1 = s_1 R_1 + s_2 R_2 \widehat{\mathbf{R}}_2 \cdot \widehat{\mathbf{R}}_1. \tag{6}$$

The second term in (6) is an error term as it depends on s_2.

Now as source s_1 is associated with a direction $\widehat{\mathbf{R}}_1$ in the phase plot, this source can be represented in the plot as a vector $\widetilde{\mathbf{s}}_1$ given by

$$\widetilde{\mathbf{s}}_1 = \widetilde{s}_1 \widehat{\mathbf{R}}_1. \tag{7}$$

This source is associated with a constant direction or heading. As $\widetilde{s}_1$ varies with time, so does the magnitude of the vector $\widetilde{\mathbf{s}}_1$.

Substituting for $\widetilde{s}_1$ from (6) into (7), this vector can be written as

$$\widetilde{\mathbf{s}}_1 = s_1 R_1 \widehat{\mathbf{R}}_1 + s_2 R_2 \left(\widehat{\mathbf{R}}_1 \cdot \widehat{\mathbf{R}}_2\right) \widehat{\mathbf{R}}_1. \tag{8}$$

Now from (5), the vector representing the phase plot is given by

$$\mathbf{z} = s_1 R_1 \widehat{\mathbf{R}}_1 + s_2 R_2 \widehat{\mathbf{R}}_2. \tag{9}$$

Comparing (8) and (9), it can be seen that the contribution from s_2 has been reduced in (8) because in general $|\widehat{\mathbf{R}}_1 \cdot \widehat{\mathbf{R}}_2| < 1$. It can also be seen in (8) and (9) that the terms in s_1 are identical. Hence, one can subtract (8) from (9) as follows:

$$\widetilde{\mathbf{s}}_2 = \mathbf{z} - \widetilde{\mathbf{s}}_1 = s_2 R_2 \left[\widehat{\mathbf{R}}_2 - \left(\widehat{\mathbf{R}}_1 \cdot \widehat{\mathbf{R}}_2\right) \widehat{\mathbf{R}}_1\right]. \tag{10}$$

and one now has an estimate of the vector representing source s_2 in the phase plot. The magnitude of this vector is equal to the underlying source s_2 to that within a multiplicative constant.

Referring again to the analogy with radar plots, one is looking for data segments where the "heading" does not change over three consecutive data points; in practice, because of the effects of noise, one would be looking for a *minimum* change in heading over these three points; in [17], this method is referred to as the minimum heading change (MHC) method and more details regarding the algorithm for this method can be found in this reference.

The MHC method is based on computing a "velocity" vector $\mathbf{v}[n]$, which is given by

$$\mathbf{v}[n] = \mathbf{z}[n] - \mathbf{z}[n-1] \tag{11}$$

at each sample point n.

In [17] the algorithm used to detect the principal directions $\{\widehat{\mathbf{R}}_i\}$ is described in detail. Also in [17] an analysis was presented for the general N source case where it was shown that the N sources could be estimated as follows:

$$\widetilde{s}_1 = C_{11}s_1 + \sum_{j=2}^{N} C_{1j}s_j, \tag{12}$$

$$\widetilde{s}_2 = C_{22}s_2 + \sum_{j=3}^{N} C_{2j}s_j, \tag{13}$$

$$\vdots$$

$$\widetilde{s}_{N-1} = C_{N-1,N-1}s_{N-1} + C_{N-1,N}s_N, \tag{14}$$

$$\widetilde{s}_N = C_{NN}s_N, \tag{15}$$

where C_{ij} is related to the cross-correlation coefficient between source i, s_i, and source j, s_j. In fact, when sources i and j are uncorrelated and prewhitening of data has taken place, it can be shown that $C_{ij} = 0$ for all i and j such that $i \neq j$; in this particular case, it can be seen from (12) to (15) that

$$\widetilde{s}_i = C_{ii}s_i, \tag{16}$$

and each source is extracted to be within a multiplicative constant.

In order to avoid the effects of spurious noise, let us write the N components of the velocity vector as

$$\mathbf{v}[n] = \left(v_1[n], v_2[n], \ldots, v_N[n]\right). \tag{17}$$

A velocity vector is accepted at sample point n if

$$V_{\max}[n] \geq v^{\text{th}} \cdot v_{\max}, \tag{18}$$

where

$$\begin{aligned} & V_{\max}[n] \\ & \quad = \max\{|v_1[n]|, |v_2[n]|, \ldots, |v_N[n]|\} \quad 0 < v^{\text{th}} < 1, \end{aligned} \tag{19}$$

$$v_{\max} = \max\{|\mathbf{v}[1]|, |\mathbf{v}[2]|, \ldots, |\mathbf{v}[M]|\} \tag{20}$$

is the maximum value of the magnitude of the velocity vector over all M sample points.

2.2. Extension to the MHC Method. In this paper, the algorithm described in [17] is extended to extracting individual sparse sources $\{s_1, s_2, \ldots, s_{N-1}\}$ from (12) to (15) for the general case where the sources are correlated. The sparsity of these sources is used to achieve this.

2.2.1. Mixtures of Two Sources. First, we will look at the case of two mixtures of two sources. From (12) and (13) with $N = 2$, it can be seen that, after applying MHC, the two estimated sources are given by

$$\tilde{s}_1 = C_{11}s_1 + C_{12}s_2, \tag{21}$$

$$\tilde{s}_2 = C_{22}s_2. \tag{22}$$

Now, if $\tilde{s}_2$ is plotted against $\tilde{s}_1$, the following vector in two-dimensional phase space results:

$$\mathbf{z} = (\tilde{s}_1, \tilde{s}_2). \tag{23}$$

Substituting for $\tilde{s}_1$ from (21) and $\tilde{s}_2$ from (22) into (23),

$$\begin{aligned} \mathbf{z} &= (C_{11}s_1 + C_{12}s_2, C_{22}s_2) \\ &= (C_{11}, 0)s_1 + (C_{12}, C_{22})s_2. \end{aligned} \tag{24}$$

Hence,

$$\mathbf{z} = s_1\mathbf{R}_1 + s_2\mathbf{R}_2, \tag{25}$$

where the vectors $\mathbf{R}_1$ and $\mathbf{R}_2$, corresponding to the underlying sources 1 and 2, respectively, are given by

$$\mathbf{R}_1 = (C_{11}, 0), \qquad \mathbf{R}_2 = (C_{12}, C_{22}). \tag{26}$$

Equation (25) can be rewritten as

$$\mathbf{z} = s_1R_1\widehat{\mathbf{R}}_1 + s_2R_2\widehat{\mathbf{R}}_2, \tag{27}$$

where $\widehat{\mathbf{R}}_1$ and $\widehat{\mathbf{R}}_2$ are unit vectors in the directions of $\mathbf{R}_1$ and $\mathbf{R}_2$, respectively.

The question now to be asked is whether one can use the fact that $\tilde{s}_2$ is proportional to s_2 only to somehow subtract the contribution from this source to $\tilde{s}_1$. Now s_2 is assumed sparse, so that s_1 will exist on its own for a segment of time which will manifest itself in the phase plot as a straight line over at least 3 points. If one assumes that s_1 is also sparse, then there will be a straight line segment corresponding to s_2 as well.

To illustrate this approach, the MHC has been applied to the data mixture in Figure 2.

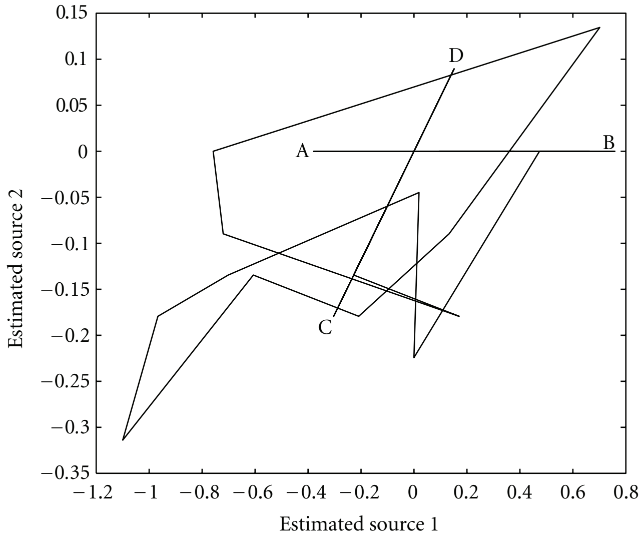


FIGURE 4: Phase plot of $\tilde{s}_2$ against $\tilde{s}_1$.

The phase plot for $\tilde{s}_2$ against $\tilde{s}_1$ for the estimated sources is given in Figure 4.

Two sets of line segments can be seen where the "heading" is constant over three or more data points: between A and B this segment corresponds to vector $\mathbf{R_1}$ in (26) corresponding to samples between 20 and 27 in Figure 1 where estimated source $\tilde{s}_1$ (corresponding to Source B in Figure 1) exists on its own; this straight line has a zero slope in phase space as estimated source $\tilde{s}_2$, corresponding to source A in Figure 1, is zero in this data segment. The trajectory between C and D in Figure 4 corresponds to samples 0 to 7 in Figure 1, where source s_2 (corresponding to source A in Figure 1) is on its own ($\mathbf{R_2}$ in (26)); from (21) and (22), this source contributes to both $\tilde{s}_1$ and $\tilde{s}_2$ so this line has a slope in phase space that is nonzero and of finite value.

Suppose that the MHC method is applied with the data inputs, this time being $\tilde{s}_1$ and $\tilde{s}_2$. Let us assume that the direction CD is detected at a certain sample point say at $n = M$; this direction can be associated with the unit vector $\widehat{\mathbf{R}}_2$ in (26).

If one takes the scalar product of the vector $\mathbf{z}$ in (27) and $\widehat{\mathbf{R}}_2$ we obtain an "estimate" of the underlying source s_2 from

$$\tilde{s}_2^{(1)} = \mathbf{z} \cdot \widehat{\mathbf{R}}_2, \tag{28}$$

where, from (26),

$$\widehat{\mathbf{R}}_2 = \frac{1}{R_2}(C_{12}, C_{22}). \tag{29}$$

Substituting for $\mathbf{z}$ from (27) into (28),

$$\tilde{s}_2^{(1)} = s_1R_1\left(\widehat{\mathbf{R}}_1 \cdot \widehat{\mathbf{R}}_2\right) + s_2R_2. \tag{30}$$

$\tilde{s}_2^{(1)}$ is not an exact estimate of s_2; this is obtained from (22), but $\tilde{s}_2^{(1)}$ will later be used to estimate source s_1.

Now the estimate of source 1 is given by (21):

$$\tilde{s}_1 = C_{11}s_1 + C_{12}s_2. \tag{31}$$

Comparing (30) with (31), an improved estimate of source s_1 can be obtained from

$$\tilde{s}_1^{(1)} = \tilde{s}_1 - \frac{C_{12}}{R_2}\tilde{s}_2^{(1)} = C' s_1, \tag{32}$$

where

$$C' = \left(C_{11} - \frac{C_{12}}{R_2}\left(R_1 \widehat{\mathbf{R}}_1 \cdot \widehat{\mathbf{R}}_2\right)\right). \tag{33}$$

Note that the subtraction in (32) can be carried out as we already know the components of $\mathbf{R}_2 = (C_{12}, C_{22})$ from the detection step and $R_2 = \sqrt{C_{12}^2 + C_{22}^2}$; prior knowledge of the coefficient C_{11} is not required. Another way of expressing (32) is to note that, from (29), C_{12} is the first component of $\widehat{\mathbf{R}}_2$:

$$C_{12} = \widehat{R}_2^{(1)}. \tag{34}$$

Using (28) and (34), (32) can be rewritten:

$$\tilde{s}_1^{(1)} = \tilde{s}_1 - \left(\mathbf{z} \cdot \widehat{\mathbf{R}}_2\right) \widehat{R}_2^{(1)}. \tag{35}$$

From (32), it can be seen that source s_1 has now been estimated to be within a scaling constant.

This procedure is like operating the MHC in "reverse" starting from the last estimated source, $\tilde{s}_2$, and then obtaining an improved estimate $\tilde{s}_1^{(1)}$ of source $\tilde{s}_1$ so this processing step is referred to as "reverse iteration." The first processing step described in Section 2.1 and [17] will be referred to as the "forward iteration."

In the Reverse processing step, it is necessary to detect segment CD in Figure 4; to avoid detecting segment AB, an *additional* test is carried out. The change in heading is calculated as described in [17], *and*, in addition, the correlation coefficient between $\tilde{s}_1$ and $\tilde{s}_2$ over P data points is computed. In order to guard against detecting segments like AB where a source exists in one component only, the minimum heading change is accepted only if the magnitude of the correlation coefficient is no smaller than a minimum threshold value $\text{Corr}_{\min}$. Once the point where this occurs is found, the heading vector is used in (35) to subtract off the contribution from s_2 to the overall phase plot. However, to avoid chance correlations by noise, the correlation is only calculated if the velocity components in phase space have a magnitude that is larger than a minimum threshold, $T_{\min}$, given by

$$T_{\min} = \lambda^{\text{th}} v_{\max}, \tag{36}$$

where $v_{\max}$ is the maximum velocity component, defined in (20), for vector $\mathbf{z}$ in the reverse iteration, and λ^{th} is a chosen threshold. Condition (36) is used instead of (18) in the reverse iteration. The following expression is used to calculate the correlation function:

$$\text{Corr}(x, y) = \frac{\sum_{i=1}^{P}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{P}(x_i - \overline{x})^2}\sqrt{\sum_{j=1}^{P}(y_j - \overline{y})^2}}, \tag{37}$$

where $\overline{x}$ and $\overline{y}$ are the mean values of x and y defined by

$$\overline{x} = \frac{1}{P}\sum_{i=1}^{P} x_i, \qquad \overline{y} = \frac{1}{P}\sum_{i=1}^{P} y_i. \tag{38}$$

It has been confirmed in simulations that if this correlation test is not carried out, then the method breaks down as the wrong source can be detected in the reverse iteration. No subtraction takes place if condition (36) is not satisfied.

It should be noted that, in many BSS methods, the first step is whitening. In this case, uncorrelated sources have characteristic directions in phase space that are orthogonal. In the MHC method, if whitening is used, uncorrelated components are extracted exactly in forward iteration only as $C_{ij} = 0$ for all i and j such that $i \neq j$ in (12) to (15); in this case, there is no need for the reverse iteration to be carried out; in [17], the Gram-Schmidt method was applied to perform the whitening. However, such a procedure, whilst useful, is not necessary for the MHC with reverse iteration, as whitening is not a required preprocessing procedure for this method. In particular, the MHC does not make any assumptions as to how many sources, if any, are correlated. In addition, centring, that is the subtraction of DC components from the data is also not necessary and can lead to undesirable effects where previously uncorrelated sources are assumed to be correlated. However, the problem with nonwhitening is that uncorrelated components cannot be individually estimated in the forward iteration. In this case, one is relying on the reverse iteration to achieve this and this could lead to incomplete separation in the reverse direction for inappropriate choice of input parameters. However, for sources that are correlated strongly enough, there is not much point in prewhitening. In subsequent implementation of the MHC method with reverse iteration, the data will not be whitened initially.

2.2.2. Extension to N Sources. The reverse processing step can be generalized to the N-source case, assuming that one has N data inputs. In particular, it can be shown that individual sources can be extracted even if there is correlation between pairs of sources. A partial analysis for the N-source case is given in Appendix A, with a pseudocode for the complete algorithm being given in Appendix B.

3. Results

3.1. Two Sources, Low Sparsity. The application of this procedure to the extraction of sources from the mixed signals in Figure 2 is now described; this is an example of two sources which are sparse over relatively few samples. The outputs from the "forward" iteration, $\tilde{s}_1$ and $\tilde{s}_2$, are taken and the reverse iteration of the MHC is applied to extract estimates $\tilde{s}_1^{(1)}$ and $\tilde{s}_2^{(1)}$ of the underlying sources s_1 and s_2, respectively. In this simulation, $v^{\text{th}} = 0.2$ (18), $\lambda^{\text{th}} = 10^{-5}$ (36), and the minimum allowed value of $\text{Corr}(x, y)$ in (37) is $\text{Corr}_{\min} = 0.9$, with $P = 3$ in (37); all these parameters were chosen by trial and error.

The figure of merit used is to correlate each estimate with each underlying source and make the association that has the largest correlation value—the CORRCOEF function in MATLAB is used to compute this parameter. The correlation coefficients, over the whole signals, between the estimated and actual sources for each methods are shown in Table 1. For comparison, the corresponding calculations are carried out using (i) the MHC method implementing just forward iteration with the Gram-Schmidt whitening [17], (ii) the Fast-ICA software Version 2.5 using the deflation method with the Gaussian nonlinearity [19] and (iii) the Clusterwise PCA method [11]. The MHC, with reverse iteration, has estimated both sources to be within a scaling constant. Exact estimation of the underlying sources has not taken place with the Fast-ICA and Clusterwise PCA methods. It should be noted that the cross-correlation between the actual sources in Figure 1 is 0.361 and is not negligible, which explains why the Fast-ICA method cannot extract any of the sources exactly for this example. The Clusterwise PCA method does not extract the sources exactly because the sources are sparse for only a few data points, so do not form obvious clusters of points. As predicted using (10), when using the MHC with just forward iteration [17], only one of the sources can be estimated exactly.

The robustness to noise for the MHC method is now investigated for the mixed data shown in Figure 2. The noise, which is modeled as Gaussian, is added to the samples of each data input *after* the mixing has taken place.

At the qth Monte Carlo run, the absolute value of the correlation coefficient is calculated between each source and estimate, over the whole data segment. An association between source and estimate is found by looking for the estimate that has highest correlation with that source.

Calculations are carried out for 2000 Monte Carlo runs and a noise standard deviation of 0.005 is used; this value corresponds to 1% of the peak value of $\mathbf{z}_1$ in Figure 2 and 0.5% of the magnitude of the peak value of $\mathbf{z}_2$. The absolute value of the correlation coefficients between the estimated and actual sources, averaged over all Monte Carlo runs, is found to be 0.991 and 0.817 for sources A and B, respectively, meaning that there is a tendency for the estimates of source A to be superior to those for source B. This can be explained as being due to the deflationary nature of the algorithm, where the various sources are estimated and subtracted one by one. For this particular set of simulations, it is found that source B tends to be detected first in the forward iteration. For those Monte Carlo runs where source B is detected first in the forward iteration, we can identify s_1 with source B and s_2 with source A, and so, neglecting noise, the estimates are given from (21) and (22) by

$$\tilde{s}_{\mathrm{B}} = C_{11} s_{\mathrm{B}} + C_{12} s_{\mathrm{A}}, \tag{39}$$

$$\tilde{s}_{\mathrm{A}} = C_{22} s_{\mathrm{A}}, \tag{40}$$

where the subscripts "A" and "B" correspond to sources A and B, respectively. It can be seen from (40) that $\tilde{s}_{\mathrm{A}}$, corresponding to source A, is estimated to be within a scaling constant. From (32), (39), and (40), source s_{B} is estimated in the reverse iteration according to

$$\tilde{s}_{\mathrm{B}}^{(1)} = \tilde{s}_{\mathrm{B}} - \frac{C_{12}}{R_{\mathrm{A}}} \tilde{s}_{\mathrm{A}}^{(1)}, \tag{41}$$

where

$$\mathbf{R}_{\mathrm{A}} = \left(C_{12}, C_{22} \right). \tag{42}$$

From (39), (40), and (41), it can be seen that the reverse iteration step requires an estimate of the heading vector $\mathbf{R}_{\mathrm{A}}$ in (42) corresponding to source A; this estimate will be affected by noise. The estimate $\tilde{s}_{\mathrm{B}}^{(1)}$ in (41) will depend on this noisy estimate of $\mathbf{R}_{\mathrm{A}}$ as well as on the estimates $\tilde{s}_{\mathrm{A}}$ and $\tilde{s}_{\mathrm{B}}$ in the forward iteration, which will both also be affected by noise; thus these combined errors will cause the estimate, $\tilde{s}_{\mathrm{B}}^{(1)}$, of source B in the reverse iteration to be more affected by noise than the estimate of source A, $\tilde{s}_{\mathrm{A}}$, in the forward iteration. This sensitivity to noise will limit the number of underlying sources that could be extracted with this method because, due to the deflationary nature of the method, the later a source is detected in the reverse iteration, then the more it will be affected by noise.

Table 1: Correlation coefficients between estimated and actual sources.

Method	Source A	Source B
Clusterwise PCA [11]	0.9994	0.6310
Fast-ICA [19]	0.9961	0.8970
MHC (Forward) [17]	1	0.9854
MHC (Reverse)	1	1

3.2. Simulations for Two Sources: High Sparsity. In Section 2, the proposed method was tested on mixtures of sources, where each source exists on its own for a relatively short segment of time compared with the whole data interval. In this section, the case where the underlying sources are sparser than the sources in Figure 1 are shown. A comparison is made between the performance of the MHC with reverse iteration and both the Clusterwise PCA and Fast-ICA methods. There are two datasets, each dataset consisting of a mixture of two periodic signals and each period consisting of Gaussian functions given as follows:

$$s_i(t) = a_i \exp\left[\frac{-\left(t - t_{0i}\right)^2}{2\sigma_i^{\,2}}\right], \tag{43}$$

where a_i is the amplitude, σ_i is the standard deviation, and t_{0i} is the time shift of the first period. This model was used in [17].

This Gaussian is repeated with a period T_i. 10 seconds of data are analyzed. The sampling frequency is set at 250 Hz.

The parameters used for the two source signals are as follows.

Source 1:

a_1 various values,

$t_{01} = 0.1$ s,

$\sigma_1 = 12.5$ ms,

$T_1 = 1$ s.

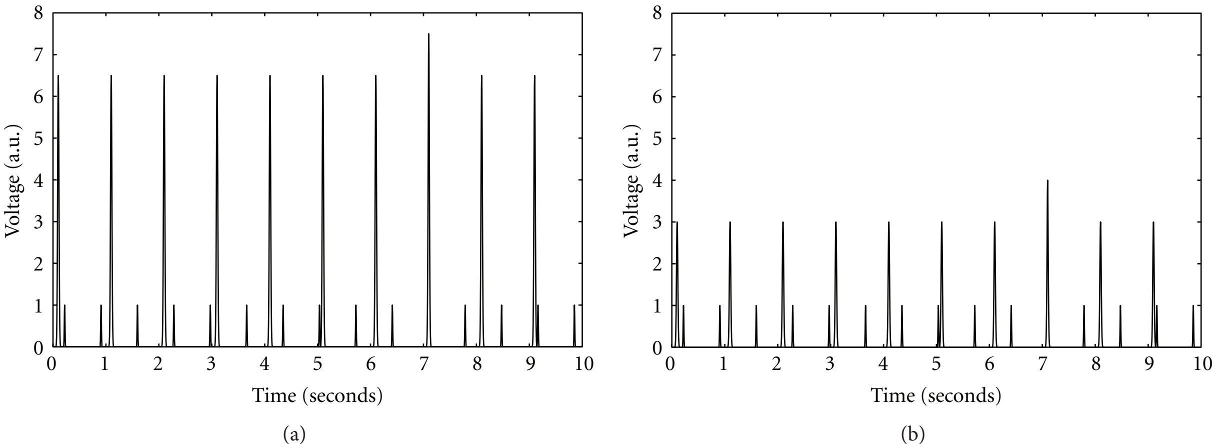


Figure 5: Simulated mixed signals.

Source 2:

a_2 = various values,

$t_{02} = 0.22$ s,

$\sigma_2 = 6.25$ ms,

$T_2 = 0.688$ s.

We now look at the case for two signals each composed of a mixture of sources 1 and 2 above with the following amplitudes:

measurement set 1: $a_1 = 6.5$, $a_2 = 1$,

measurement set 2: $a_1 = 3$, $a_2 = 1$,

$N = 2$ in (1).

The two mixed signals are shown in Figure 5.

These signals could represent, for example, a simplified model of the ECG measurements taken from the abdomen of an expectant mother, where source 1 represents a contribution from the maternal ECG and source 2 a contribution from the fetal ECG. In this paper, this dataset is separated into two segments: segment 1 where there is no coincidence between the two sources, between 0.5 s and 2.5 s in Figure 5, and segment 2 where coincidence between the sources occurs, between 7 s and 9 s in Figure 5—this could, for example, represent the maternal/fetal coincidence in abdominal ECG data.

The following parameters in the MHC are chosen after some experimentation: $\text{Corr}_{\min} = 0.999$, $\lambda^{\text{th}} = 10^{-5}$ and $\upsilon^{\text{th}} = 0.5$. Also, in the reverse iteration, it is found that best results are obtained by using $P = 5$ in (37).

The Fast-ICA method is run using the deflation method with the Gaussian nonlinearity. The parameters used in the Clusterwise PCA method are the same as listed in [11].

The various methods are assessed by computing the correlation coefficient between the estimated and actual sources using the same methodology used for the analysis in Table 1.

The results of applying the Clusterwise PCA, Fast-ICA, and MHC (reverse) methods to segments 1 and 2 are shown in Table 2.

Table 2: Correlation between estimate and source for (a) Segment 1 and (b) Segment 2.

(a)

Method	Source 1	Source 2
MHC (Reverse)	1	1
Clusterwise PCA	1	1
Fast-ICA	1	0.9993

(b)

Method	Source 1	Source 2
MHC (Reverse)	1	1
Clusterwise PCA	0.6492	0.9993
Fast-ICA	0.9479	0.9997

In Table 2, it can be seen that for segment 1, both the Clusterwise PCA and MHC reverse iteration methods are able to extract the sources exactly. The Fast-ICA method subtracts the average from the original data and this introduces correlation between the assumed sources and so the correlation coefficients between estimated and actual sources are less than 1 for one of the sources. For segment 2, only the MHC method with reverse iteration manages to extract the two sources to be within a scaling constant. The Fast-ICA cannot extract the sources exactly because there is now significant correlation between the two underlying sources due to the coincidence occurring between the two sources. The Clusterwise PCA method also breaks down when estimating source 1 because of the overlap between sources 1 and 2; to see this further, in Figure 6, the phase plot of the data for segment 2 is shown: 1 and 2 refer to the contributions to the phase plot for sources 1 and 2, respectively. "12" is the contribution to the phase plot from the overlapping of sources 1 and 2 that occurs in segment 2; this feature is of significant size compared with the features representing source 1 only and source 2 only. The Clusterwise PCA method attempts to group the plot into two "clusters."

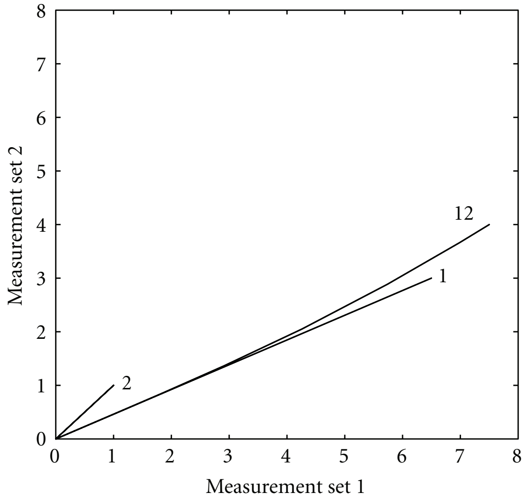


FIGURE 6: Phase plot for segment 2. "1" and "2" refer to intervals in the phase plot corresponding to sources 1 and 2, respectively, and the label "12" refers to coincidence between the two sources.

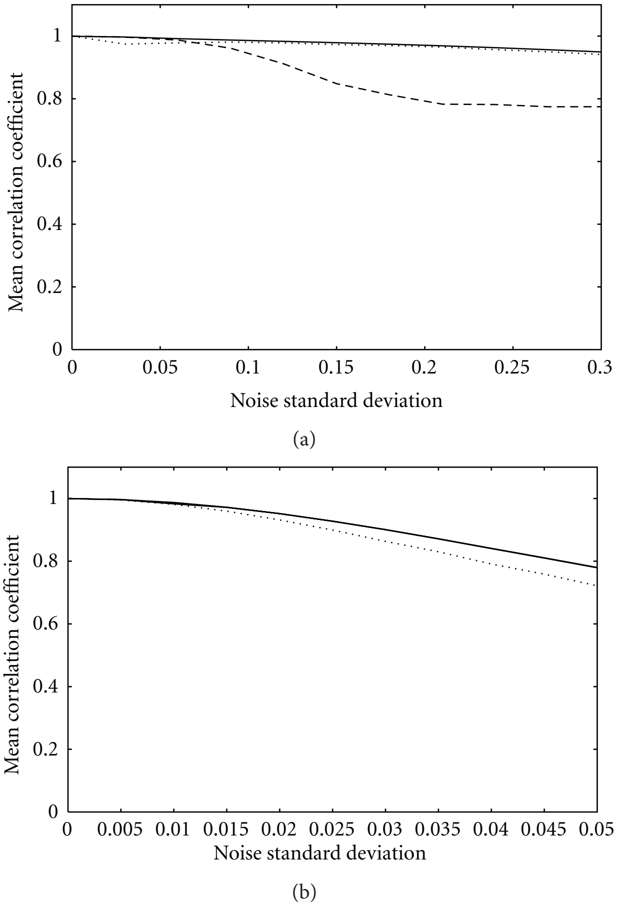


FIGURE 7: Mean correlation coefficient as a function of noise standard deviation: (a) source 1 (b) and source 2. Full line: Fast-ICA, dashed line, Clusterwise PCA; dotted line: MHC (reverse).

The contribution 12 is clustered with 1, which degrades the estimate of this source. However, this contribution has less of an effect on the cluster corresponding to source 2, and hence this source is estimated much better, as observed in Table 2.

The robustness to the noise of the proposed method is now looked for segment 1 where there are no coincidences between the two underlying sources. The amplitude of source 2 for each set of data is equal to 1; the Gaussian noise is added with standard deviations varying from 0 to 30% of the amplitude of source 2. 2000 Monte Carlo runs are used.

The absolute value of the correlation coefficient between each source and closest estimate is averaged over all Monte Carlo runs. A comparison is made between the performances of the proposed method, the Fast-ICA and the Clusterwise PCA methods. In Figure 7(a), the mean correlation is plotted as a function of the standard deviation of noise for source 1, and in Figure 7(b) the corresponding plot is shown for source 2. Note that the range of noise standard deviations used in the plots are different as the extraction of the smaller source 2 breaks down at smaller noise levels.

In Figure 7(a), it can be seen that MHC and Fast-ICA have comparable performances when estimating source 1, with the latter method having slightly more robustness to noise. The Clusterwise PCA method is more sensitive to noise, particularly for noise standard deviation larger than 0.2; the problem here is that if the noise is too large, then points in phase space will be associated with the wrong cluster. All the methods give a comparable performance when estimating source 1 for noise standard deviations up to 0.05, corresponding to 5% of the amplitude of source 2 in each set of data.

In Figure 7(b), when estimating source 2, Fast-ICA and Clusterwise PCA have indistinguishable performances for the range of standard deviations of noise used (0 to 5% of the amplitude of source 2). MHC is more sensitive to noise than the other two methods for noise standard deviations greater than about 1% of the amplitude of source 2. The reason for this is that Fast-ICA and Clusterwise PCA process the data globally, whilst MHC does this locally, so is more susceptible to noise.

Now it is of interest to ask whether, when using the MHC, the reverse iteration method is more susceptible to noise than if the forward estimation only is used. In Figure 8, the mean correlation coefficient between estimate and source for the forward MHC with no whitening is compared with MHC with the reverse iteration for source 2. We can see that reverse iteration does not contribute significantly more errors to the estimation of source 2. Hence most of the errors occurring in the MHC reverse method are occurring in the forward iteration.

For this particular simulation, the mixing matrix is given by $A_1 = \left(\begin{smallmatrix} 6.5 & 1 \\ 3 & 1 \end{smallmatrix}\right)$ and the phase space plot for the mixed signals is shown in Figure 6. The directions corresponding to sources 1 and 2 are very clear in the directions $(6.5, 3)$ and $(1, 1)$. However, a much stricter test for the reverse MHC method is

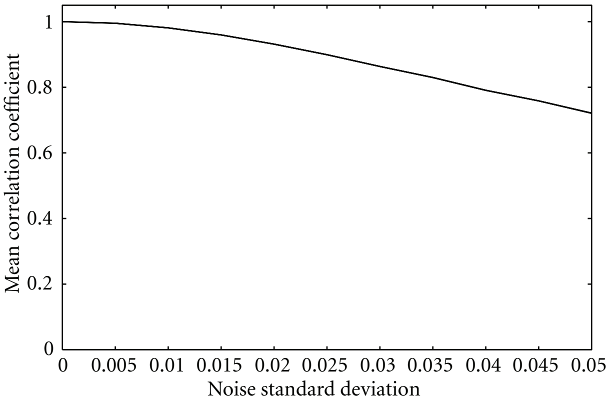


Figure 8: Extraction of source 2: full line: MHC with reverse iteration, dashed line: MHC with forward iteration only.

when these two directions are much closer. Simulations have also been carried out for the following mixing matrices:

$$A_2 = \begin{pmatrix} 3.2 & 1 \\ 3 & 1 \end{pmatrix},$$
$$A_3 = \begin{pmatrix} 3.02 & 1 \\ 3 & 1 \end{pmatrix} \tag{44}$$

for segments of time 0.5 to 2.5 seconds (no coincidence of sources 1 and 2) and 7 s to 9 s where there is coincidence. For 0.5 to 2.5 seconds, it is found that all three methods tested have a similar performance as when using A_1, namely that Clusterwise PCA and MHC reverse extracts the sources exactly, and Fast-ICA extracts the sources with correlation coefficients 0.999 for source 1 and 1 for source 2. A stricter test of the performance of these methods when using mixing matrices A_2 and A_3 is to look at the segment of data, where there is coincidence of sources between 7 s and 9 s. The performances of these methods for this data segment are shown in Tables 3 and 4.

The MHC (reverse) method still gives perfect extraction of sources to be within a scaling constant and has the best performance of the three methods tested for this particular dataset.

3.3. Two Sources, Varying Sparsity. It is of interest to investigate the performance of the proposed algorithm when processing mixtures of sparse sources with different "sparsity." At one extreme, one has sources that completely overlap and where there are no segments where one source exists on its own. At the other extreme, one could have sources that do not overlap. An intermediate situation is where there is partial overlap between sources.

In the next set of simulations, two mixtures of two sources are shown, where each source is generated from a uniform random number generator with values between −1 and +1; each source consists of 100 samples.

Table 3: Correlation between estimate and source for mixture matrix A_2.

Method	Source 1	Source 2
MHC (Reverse)	1	1
Clusterwise PCA	0.7970	0.9894
Fast-ICA	0.9480	0.9997

Table 4: Correlation between estimate and source for mixture matrix A_3.

Method	Source 1	Source 2
MHC (Reverse)	1	1
Clusterwise PCA	0.7133	0.9992
Fast-ICA	0.9480	0.9997

The two sources are mixed using the randomly selected mixing matrix:

$$A = \begin{pmatrix} 0.574580785 & 0.166294188 \\ -0.668082703 & -0.933171682 \end{pmatrix}. \tag{45}$$

Next, the second source is shifted by L samples with respect to the first source and the mixing matrix (45) applied to the two sources to produce mixtures with $100 + L$ samples. For each value of L, the MHC algorithm with reverse iteration is applied; the following parameters are found empirically to work for this data: $\upsilon^{\text{th}} = 0.1$ (18), $\lambda^{\text{th}} = 10^{-5}$ (36), and the minimum allowed value of $\text{Corr}(x, y)$ in (37) is $\text{Corr}_{\min} = 0.9$, with $P = 3$ in (37).

In Figure 9, the correlation coefficient between estimated sources and closest actual sources, for sources 1 and 2, is plotted as a function of L for the Fast ICA, Clusterwise PCA, and MHC reverse methods.

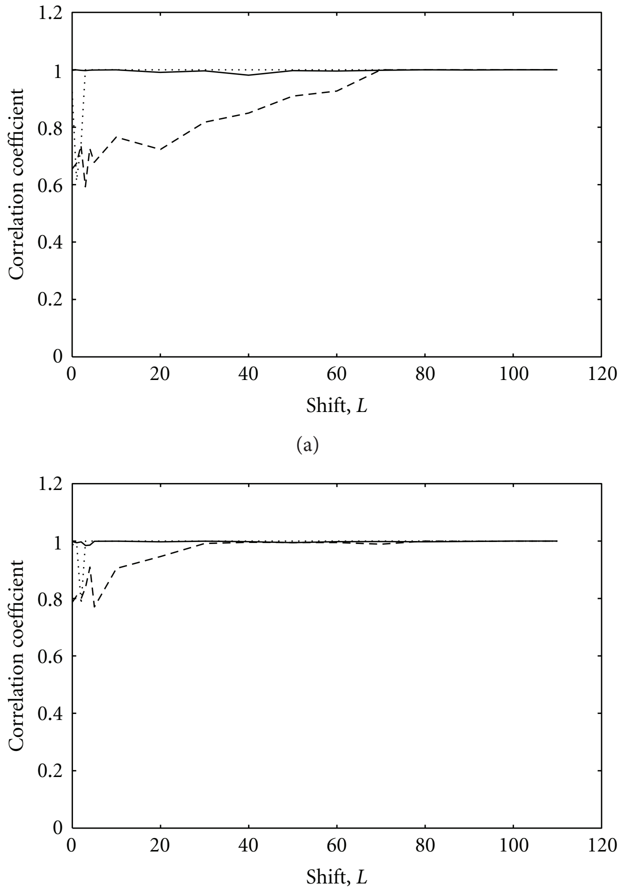
The following observations can be made.

(i) Clusterwise PCA extracts both sources with correlation coefficients larger than 0.98 for $L \geq 70$.

(ii) Fast-ICA should be expected to work best for $L = 0$ and L around 100 because in these two extremes the sources are uncorrelated. At some intermediate values of L,where there is some correlation, it is expected that Fast-ICA should not work so well, and this is observed in this example for L between 30 and 50 for source 1. However, the estimates for source 2 have correlation coefficients larger than 0.98 for all values of L.

(iii) Fast-ICA has the best overall performance for all values of L. However, if $L \geq 3$, then MHC (reverse) has the best performance as this method requires only three samples to work; Clusterwise PCA requires larger values of L to work because it performs a global clustering of the data.

(iv) The Fast-ICA algorithm is based on iteratively updating weights applied to the data. The initial values of these weights are chosen randomly; hence, one can obtain different results by rerunning the Fast-ICA algorithm and it is found that reinitializing two or more times can lead to better results for some values



(a)

(b)

Figure 9: Mean correlation coefficient as a function of shift, L, between sources: (a) source 1 and (b) source 2. Full line: Fast-ICA; dashed line, Clusterwise PCA; dotted line: MHC (reverse).

of L. When using Fast ICA, the symmetric approach is found to work better than deflation for these signals. Changing the nonlinearity from gauss to pow 3 and sometimes changing the step size in the iteration of weights can lead to better results; in Figure 9, the best estimates, changing the input parameters, are used to construct the performance curves for Fast ICA. So, for this mixture of sources, the estimation using Fast-ICA can be very sensitive to the approaches used and to input parameters; however, for MHC (reverse), the same set of parameters works for all values of $L \geq 3$.

3.4. 4-Lead ECG Data. In order to compare the performances on actual data, of the Fast-ICA, MHC, and Clusterwise PCA methods, data taken from the Daisy Database [20] will be analyzed. These data consists of ECG signals taken from an expectant mother. The data consists of 8 leads, 1 to 5 being abdominal and 6 to 8 thoracic. There are 2500 samples of data; there is some uncertainty about the sampling frequency, also pointed out in [10], but it is probably 250 Hz.

These data were analysed in [17], where the MHC with forward iteration was applied to the data. Successful extractions of the fetal and maternal signals were achieved.

In this paper, the MHC with reverse iteration is also applied. 4 leads are chosen for analysis: leads 1, 2, 3, and 4—this is a more challenging problem in extracting the fetal signal than when using 8 leads. In [17], all the data samples are analyzed; in this paper, data used is between sample numbers 1451 to 1751 where there is an overlap between a maternal and fetal complex as shown in Figure 10(a) where the maternal and fetal R-waves are marked by "M" and "F," respectively.

A comparison is made between the Fast-ICA (deflation, Gaussian), MHC forward iteration (with the Gram-Schmidt initialization), MHC with reverse iteration with no whitening of data, and Clusterwise PCA. The input parameters for the MHC calculations were $\upsilon^{\text{th}} = 0.5$ in (18), $\text{Corr}_{\text{min}} = 0.999$, $P = 5$ in (37), and $\lambda^{\text{th}} = 10^{-5}$ in (36).

For each method applied, the output that is closest to the expected fetal signal is chosen. The results for the Fast-ICA, MHC (forward), MHC (reverse), and Clusterwise PCA methods are shown in Figures 10(b), 10(c), 10(d), and 10(e), respectively.

It can be seen that all four methods give comparable results for the extraction of the fetal sources.

Comparing Figures 10(c) and 10(d), it is observed that the MHC with forward iteration gives similar results for the fetal source as the MHC with reverse iteration. This is due to the fact that despite the maternal/fetal coincidence in the data, there is still low correlation between the maternal and fetal sources. Hence when whitening is carried out, the vectors representing the maternal and fetal sources in the resulting phase plot are almost perpendicular and, from (16), this means that the extracted sources in the forward iteration are approximately a scaled version of the actual sources. The MHC with reverse iteration does not whiten the data initially but is still able to extract the fetal source with comparable quality to the other three methods applied to this data. It is interesting to note that noise is not having a deleterious effect on the quality of the estimated fetal sources in Figures 10(b), 10(c), 10(d), and 10(e); for this particular set of data, the anti-aliasing filter has already filtered off a significant amount of noise.

4. Discussion and Conclusions

The MHC method looks locally for three *consecutive* points that are the most linear of all points in phase space; hence just *one* direction corresponding to one source is estimated at any given time. Conventional clustering methods estimate either the mixing or the separating matrices and some algorithms can be extended to the underdetermined case, which cannot be done at present with the MHC. The MHC is a more straightforward method to implement with no iterative clustering techniques; however this technique requires the sources to be strictly sparse whilst clustering techniques can deal with sources that are approximately sparse where one looks for clustering of points in phase space around a straight line [11]. As the MHC analyzes data in phase space locally,

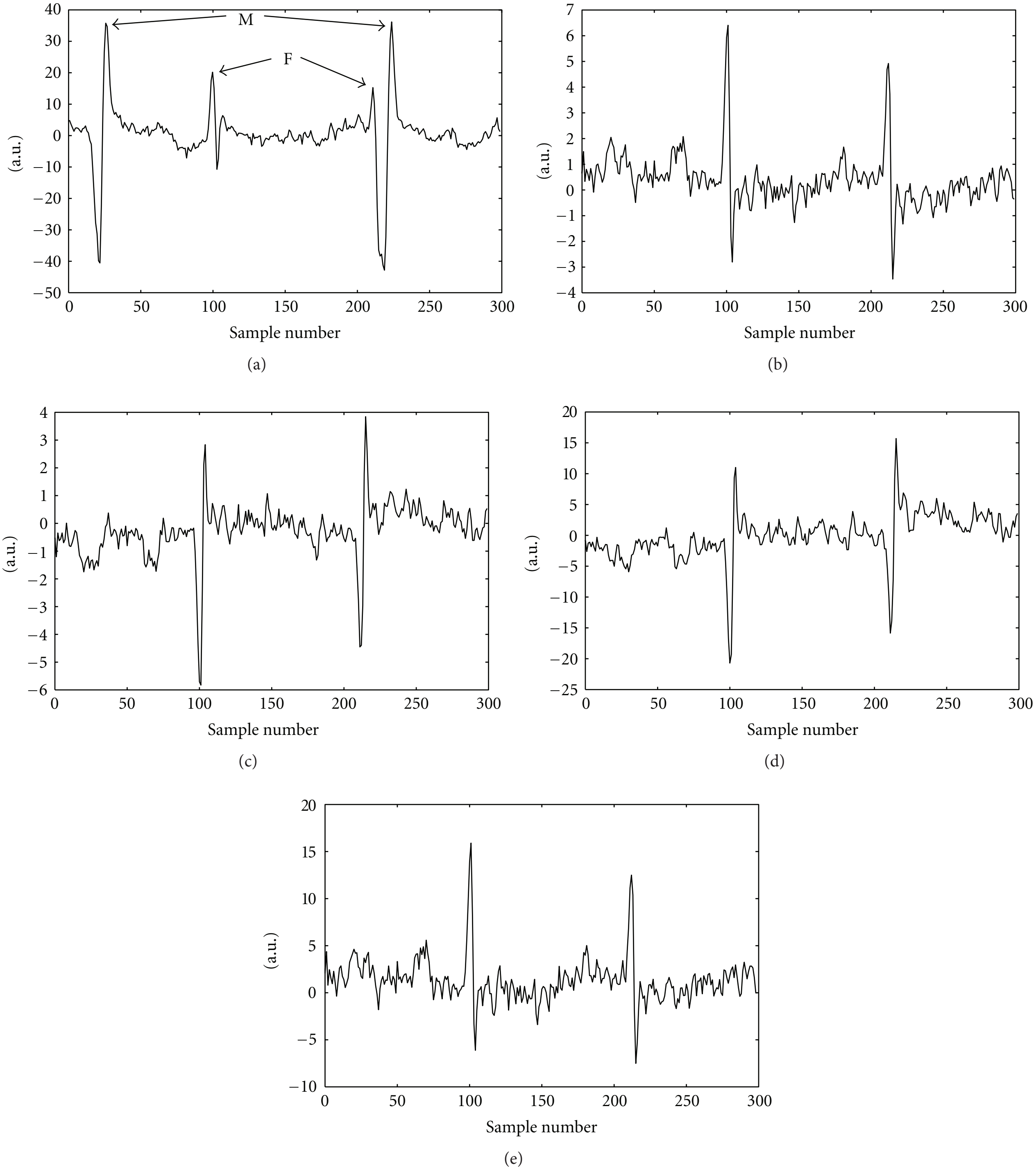


Figure 10: Abdominal and thoracic ECG data [20]. (a) Lead 1, (b) extraction of fetal ECG using Fast-ICA, (c) extraction of fetal ECG using MHC (forward iteration, Gram-Schmidt's initialization), (d) Extraction of fetal ECG using MHC (Reverse iteration), (e) Clusterwise PCA.

it would be more sensitive to the effects of noise than the clustering techniques. For the MHC to work, the underlying sources need to be sparse for only a few (down to three data points) which can be seen as an advantage of this method over methods that look for the clustering of data points along a line in phase space, where it is assumed that the underlying sources are sparse over several segments of the mixed data.

The MHC reverse iteration method can be sensitive to the choice of input parameters υ^{th} in (18), λ^{th} in (36), Corr_{min}, and P in (37). On the other hand, both the Fast-ICA and Clusterwise PCA algorithms have been found to be relatively insensitive to the choice of the input parameters. This sensitivity to input parameters implies that, in its present form, the reverse MHC method should be used to analyse signals offline, rather than online, where the parameters can be varied to obtain optimal estimation of the underlying sources.

It should be noted that MHC relies on *all* sources being sparse. The more the number of sources, the less likely this assumption will be obeyed. In addition, as discussed in Section 3.1, the sensitivity to the noise of the method limits the number of sources that could be extracted. One potential application of the MHC method is the analysis of electron paramagnetic resonance (EPR) data [5] to detect the presence of free radicals in chemical and biological samples where there are two mixtures of two sources to be analyzed. In [5] Fast-ICA is applied to two sets of measured spectra to extract the spectra corresponding to the individual radicals. The results found by authors are promising. However, the underlying source spectra (e.g., for OH and O_2) consist of overlapping peaks, and hence there is some correlation between the underlying sources. In addition, the underlying spectra can be modeled as sparse as there are distinct peaks in the data. It would be of interest to investigate whether the MHC method could also be used in this application and whether it would give any improved results compared with using Fast-ICA.

There are BSS methods existing in the literature that derive methods to estimate the underlying sources where the underlying sources are assumed to be correlated. For example, in [21], a method based on using precoders is described; the algorithm to estimate the sources is more complicated than the method proposed in this paper, but the former method does not assume that the underlying sources are sparse. In [22] it is assumed that the underlying sources are nonnegative, and, as for [21], the sources are not assumed to be sparse. It would be of interest to compare the method proposed in this paper with the algorithms in [21, 22] for mixtures of sparse, correlated sources.

Appendices

A. Reverse Processing for N-Sources

From (15), the last estimated source is equal to the actual source to within a multiplicative constant:

$$\tilde{s}_N \propto s_N, \tag{A.1}$$

which can be rewritten as

$$\tilde{s}_N = D_N s_N, \tag{A.2}$$

where for notational convenience we have put $D_N = C_{NN}$ with C_{NN} given in (15).

Hence (12) to (15) can be rewritten as

$$\tilde{s}_1 = C_{11}s_1 + \sum_{j=2}^{N} C_{1j}s_j, \tag{A.3}$$

$$\tilde{s}_2 = C_{22}s_2 + \sum_{j=3}^{N} C_{2j}s_j, \tag{A.4}$$

$$\vdots$$

$$\tilde{s}_{N-1} = C_{N-1,N-1}s_{N-1} + C_{N-1,N}s_N, \tag{A.5}$$

$$\tilde{s}_N = D_N s_N. \tag{A.6}$$

Note that $\tilde{s}_{N-1}$ depends on the actual sources s_{N-1} and s_N whilst $\tilde{s}_N$ depends on s_N only. Hence the reverse processing step can be applied using the pair of estimated sources $\tilde{s}_{N-1}$ and $\tilde{s}_N$ as the inputs. From the theory described in Section 2.2.1, after subtracting off the contribution from s_N to $\tilde{s}_{N-1}$, an improved estimate, $\tilde{s}_{N-1}^{(1)}$ of s_{N-1} is obtained that depends on s_{N-1} only; that is, one can write:

$$\tilde{s}_{N-1}^{(1)} = D_{N-1}s_{N-1}. \tag{A.7}$$

The application of the reverse processing iteration with $\tilde{s}_{N-1}$ and $\tilde{s}_N$ as the inputs and with $\tilde{s}_{N-1}^{(1)}$ as the output is written using the following notation:

$$\tilde{s}_{N-1}^{(1)} = R\left(\tilde{s}_{N-1}, \tilde{s}_N\right) \tag{A.8}$$

with "R" representing the operation of reverse processing.

Hence, replacing $\tilde{s}_{N-1}$ in (A.5) by $\tilde{s}_{N-1}^{(1)}$ in (A.8), (A.3) to (A.6) can be replaced by the following set of equations:

$$\tilde{s}_1 = C_{11}s_1 + \sum_{j=2}^{N} C_{1j}s_j, \tag{A.9}$$

$$\tilde{s}_2 = C_{22}s_2 + \sum_{j=3}^{N} C_{2j}s_j, \tag{A.10}$$

$$\vdots$$

$$\tilde{s}_{N-2} = C_{N-2,N-2}s_{N-2} + C_{N-2,N-1}s_{N-1} + C_{N-2,N}s_N, \tag{A.11}$$

$$\tilde{s}_{N-1}^{(1)} = D_{N-1}s_{N-1}, \tag{A.12}$$

$$\tilde{s}_N = D_N s_N. \tag{A.13}$$

Equations (A.11), (A.12), and (A.13) are used to estimate s_{N-2}. First a decision needs to be made as to which of $\tilde{s}_{N-1}^{(1)}$ (A.12) and $\tilde{s}_N$ (A.13) to subtract off the corresponding sources from $\tilde{s}_{N-2}$ in (A.11). The procedure is as follows. First the MHC procedure in reverse is applied to $\tilde{s}_{N-1}^{(1)}$, and $\tilde{s}_{N-2}$ and the magnitude of the minimum heading change, $\varepsilon_{N-1,N-2}$ is found subject to the correlation coefficient over the P data points being at least $\text{Corr}_{\min}$, (37). In principle, the minimum heading change should occur when s_{N-1} exists on its own and ideally $\varepsilon_{N-1,N-2} = 0$. However, a situation could occur that s_{N-1} is somehow "hidden" by the other two sources in that it does not reveal itself as a straight line of the phase plot of the original data. Subsequently, the MHC procedure in reverse is applied to $\tilde{s}_N$ and $\tilde{s}_{N-2}$ and the magnitude of the minimum heading change, $\varepsilon_{N,N-2}$, is found for this pair of sources. The source estimate that is subtracted is determined from the minimum of $\varepsilon_{N-1,N-2}$ and $\varepsilon_{N,N-2}$. Suppose it is found that $\varepsilon_{N,N-2} < \varepsilon_{N-1,N-2}$. In this case, one performs a subtraction procedure on $\tilde{s}_N$ and $\tilde{s}_{N-2}$ as follows:

$$\tilde{s}_{N-2}^{(1)} = R\left(\tilde{s}_{N-2}, \tilde{s}_N\right). \tag{A.14}$$

```
Step 1
$(\{\tilde{s}_1\}, \{\tilde{s}_2\}, \ldots, \{\tilde{s}_N\}) = \text{MHCForward}(\{\mathbf{z}\}, v^{\text{th}})$    Section 2.1
Step 2
for $m = 1 : M$
        $\tilde{s}_N^{(0)}[m] = \tilde{s}_N[m]$    (A.6)
end
$[\{\tilde{s}_{N-1}^{(1)}\}, \Delta H, \text{NoSub}] = \text{MHCReverse}(\{\tilde{s}_{N-1}\}, \{\tilde{s}_N\}, \lambda^{\text{th}}, \text{Corr}_{\min}, P)$    (A.8)
for $i = 2 : N - 1$
        for $j = 1 : N$
                flagsub($j$) = 0
        end
        $k = 0$
        for $m = 1 : M$
                $\{\tilde{s}_{N-i}^{(0)}[m]\} = \{\tilde{s}_{N-i}[m]\}$
      end
      while $k \le i$
            $k = k + 1$
            for $j = 0 : i - 1$
                    $\Delta H(j) = 1e10$
            end
            for $j = 0 : i - 1$
                    if flagsub($j$) = 0
                          $[\{t_{N-i}^{(j)}\}, \Delta H(j), \text{NoSub}(j)] =$
                          $\text{MHCReverse}(\{\tilde{s}_{N-i}^{(k-1)}\}, \{\tilde{s}_{N-j}^{(j)}\}, \lambda^{\text{th}}, \text{Corr}_{\min}, P)$
                    end
            end
Step 3
            if min(NoSub($j$)) > 0
                $j_{\min}$ = value of $j$ : $\Delta H(j)$ is a minimum
                for $m = 1 : M$
                        $\tilde{s}_{N-i}^{(k)}[m] = t_{N-i}^{(j_{\min})}[m]$
                end
                flagsub($j_{\min}$) = 1
            else
                $k = i$
                for $m = 1 : M$
                        $\tilde{s}_{N-i}^{(i)}[m] = \tilde{s}_{N-i}[m]$
                end
            end
      end
end
```

Pseudocode 1

One can then write $\tilde{s}_{N-2}^{(1)}$:

$$\tilde{s}_{N-2}^{(1)} = C'_{N-2,N-2} s_{N-2} + C'_{N-2,N-1} s_{N-1}. \tag{A.15}$$

Next $\tilde{s}_{N-2}^{(1)}$ in (A.15) and $\tilde{s}_{N-1}^{(1)}$ in (A.12) are processed together to subtract off the contribution of s_{N-1} to $\tilde{s}_{N-2}^{(1)}$:

$$\tilde{s}_{N-2}^{(2)} = R\left(\tilde{s}_{N-2}^{(1)}, \tilde{s}_{N-1}^{(1)}\right). \tag{A.16}$$

In theory, $\tilde{s}_{N-2}^{(2)}$ will now depend on s_{N-2} only and one can then write that

$$\tilde{s}_{N-2}^{(2)} = D_{N-2} s_{N-2}. \tag{A.17}$$

This process is iterated, next calculating an estimate, $\tilde{s}_{N-3}^{(3)}$ until one has estimated all the sources $\{\tilde{s}_{N-p}^{(p)}\}$ which should, in theory, be scaled versions of the actual sources.

B. Pseudocode for Algorithm

Notes:

(1) For a scalar quantity, the notation $\{a\}$ is used to indicate the set of samples from $m = 1$ to M:

$$\{a\} = \{a[1], a[2], \ldots, a[M]\}. \tag{B.1}$$

```
function[$(\{\tilde{s}_1\}, \{\tilde{s}_2\}, \ldots, \{\tilde{s}_N\})$] = MHCForward($\{\mathbf{z}\}, v^{\text{th}}$)
Step 1
for $k = 1 : N$
        for $m = 2 : M$
                $\mathbf{v}[m] = \mathbf{z}[m] - \mathbf{z}[m-1]$
                $\hat{\mathbf{v}}[m] = \mathbf{v}[m] / |\mathbf{v}[m]|$
        end
Step 2
        for $m = 2 : M$
                $V_{\max}[m] = \max(|v_1[m]|, |v_2[m]|, \ldots, |v_N[m]|)$   (19)
        end
        $v_{\max} = \max\{|\mathbf{v}[1]|, |\mathbf{v}[2]|, \ldots, |\mathbf{v}[M]|\}$   (20)
Step3
        for $i = 1 : M-1$
                $j = i + 1$;
                if $V_{\max}[j] > v^{\text{th}} v_{\max}$
                        $\mathbf{d}^-[j] = \hat{\mathbf{v}}[i] - \hat{\mathbf{v}}[j]$
                        $\mathbf{d}^+[j] = \hat{\mathbf{v}}[i] + \hat{\mathbf{v}}[j]$
                        $\Delta[j] = \min(|\mathbf{d}^-[j]|, |\mathbf{d}^+[j]|)$
                end
        end
        $j_{\min}$ = value of $j$ where $\Delta[j]$ is a minimum
        $\widehat{\mathbf{R}}_k = \hat{\mathbf{v}}[j_{\min}]$
Step 4
        for $m = 1 : M$
                $\tilde{s}_k[m] = \mathbf{z}[m] \cdot \widehat{\mathbf{R}}_k$   (6)
                $\tilde{\mathbf{s}}_k[m] = \tilde{s}_k[m]\,\widehat{\mathbf{R}}_k$   (7)
                $\mathbf{z}[m] \leftarrow \mathbf{z}[m] - \tilde{\mathbf{s}}_k[m]$   (10)
        end
end
```

Pseudocode 2

(2) For a vector quantity, the notation $\{\mathbf{a}\}$ is used to indicate the set of samples from $m = 1$ to M of this vector quantity:

$$\{\mathbf{a}\} = \{\mathbf{a}[1], \mathbf{a}[2], \ldots, \mathbf{a}[M]\}. \qquad \text{(B.2)}$$

(3) There are three listings: the main program and two functions; for each listing, the processing steps are described first and then the listing is given with the step numbers indicated at the appropriate places.

Inputs. We have the following Inputs:

v^{th} (18),

λ^{th} (36),

$\text{Corr}_{\min}$ (37),

P (37),

$\{\mathbf{z}\}$=$(\mathbf{z}[1], \mathbf{z}[2], \ldots, \mathbf{z}[M])$ where M = number of data points.

Outputs. Estimated sources $\{\tilde{s}_{N-p}^{(p)}\}$, $p = 0, 1, 2, \ldots, N-1$ (see Appendix A).

Main Program (see Pseudocode 1)

Steps

Step 1. Estimated sources from the forward iteration using MHC.

Step 2. Estimated sources from the reverse iteration.

NoSub = 1 if a subtraction has taken place in the reverse iteration, NoSub = 0 if not; this is passed to the main routine.

i is index number for source estimate from forward iteration.

j is index number for source estimate from reverse iteration.

For subtraction from source from forward iteration.

$t_{N-i}^{(j)}$ is the estimated source in the reverse iteration when the inputs are $\tilde{s}_{N-i}^{(k-1)}$ and $\tilde{s}_{N-j}^{(j)}$ with $\Delta H(j)$ being the associated minimum heading change.

```
function$\left[\left\{\tilde{s}_{\mathrm{A}}^{(1)}\right\}, \Delta H, \mathrm{NoSub}\right] = \mathrm{MHCReverse}\left(\left\{\tilde{s}_{\mathrm{A}}\right\}, \left\{\tilde{s}_{\mathrm{B}}\right\}, \lambda^{\mathrm{th}}, \mathrm{Corr}_{\min}, P\right)$
Step 1
for $m = 1 : M$
        $\mathbf{z}[m] = \left(\tilde{s}_{\mathrm{A}}[m], \tilde{s}_{\mathrm{B}}[m]\right)$
end
for $m = 2 : M$
        $\mathbf{v}[m] = \mathbf{z}[m] - \mathbf{z}[m-1]$     Compute velocity vector, (11)
end
Step 2
for $m = 2 : M$
        $V_{\max}[m] = \max\left(\left|v_1[m]\right|, \left|v_2[m]\right|, \ldots, \left|v_N[m]\right|\right)$     (19)
end
$v_{\max} = \max\{|\mathbf{v}[1]|, |\mathbf{v}[2]|, \ldots, |\mathbf{v}[M]|\}$     (20)
Step 3
for $i = P : M$
        for $j = 1 : P$
                $\mathbf{z}^{\mathrm{local}}[j] = \mathbf{z}[i - P + j]$
        end
        $C[i] = \mathrm{corrcoef}(z_1^{\mathrm{local}}, z_2^{\mathrm{local}})$
end
Step 4
for $i = P : M - 1$
        $j = i + 1;$
        if $\left|v_1[i]\right| > \lambda^{\mathrm{th}} v_{\max}$ & $\left|v_2[i]\right| > \lambda^{\mathrm{th}} v_{\max}$ & $\left|v_1[j]\right| > \lambda^{\mathrm{th}} v_{\max}$ & $\left|v_2[j]\right| > \lambda^{\mathrm{th}} v_{\max}$
        & $|C[i]| > \mathrm{Corr}_{\min}$
                Flag = 1
                $\mathbf{d}^{-}[j] = \hat{\mathbf{v}}[i] - \hat{\mathbf{v}}[j]$
                $\mathbf{d}^{+}[j] = \hat{\mathbf{v}}[i] + \hat{\mathbf{v}}[j]$
                $\Delta[j] = \min\left(\left|\mathbf{d}^{-}[j]\right|, \left|\mathbf{d}^{+}[j]\right|\right)$
        else
                Flag = 0
        end
end
if Flag = 1
        $\Delta H = \min(\Delta[j])$
        $j_{\min}$ = value of $j$ where $\Delta[j]$ is a minimum
        $\widehat{\mathbf{R}}_{\mathrm{B}} = \hat{\mathbf{v}}[j_{\min}]$
end
Step 5
if Flag = 1
    NoSub = 1
      for $m = 1 : M$
          $\tilde{s}_{\mathrm{A}}^{(1)}[m] = \tilde{s}_{\mathrm{A}}[m] - (\mathbf{z}[m] \cdot \widehat{\mathbf{R}}_{\mathrm{B}})\widehat{R}_{\mathrm{B}}^{(1)}$     (35)
      end
else
    NoSub = 0
    for $m = 1 : M$
          $\tilde{s}_{\mathrm{A}}^{(1)}[m] = \tilde{s}_{\mathrm{A}}[m]$
    end
end
```

Pseudocode 3

Step 3. If the min(NoSub(j)) $>$ 0 for some j, then at least one of the estimated source $\widetilde{s}_{N-j}^{(j)}$ can be considered for subtraction. Choose source $t_{N-i}^{(j_{\min})}$, where the heading change is a minimum then put flagsub($j_{\min}$) = 1 to indicate that $\widetilde{s}_{N-j}^{(j)}$ should no longer be used in the subtraction process for this iteration. Otherwise put reverse estimate = forward estimate and exit loop by putting $k = i$.

End of Main Program.

Function to Carry out Forward Iteration (see Pseudocode 2)

Step 1. Loop over number of data inputs, N, and number of data points M to compute normalized headings from the velocity vector, (11).

Step 2. Determine maximum velocity to eliminate velocity vectors that have all components below a certain threshold.

Step 3. Apply MHC to look for minimum differences between adjacent headings. $\mathbf{d}^-$ and $\mathbf{d}^+$, are used to do this; $\mathbf{d}^+$ which is the sum of adjacent normalized velocity vectors takes into account headings that double back on themselves, corresponding to a peak in the data.

Step 4. Compute estimated kth source, (6) and vector in phase space representing estimated source, (7); subtract this vector from the measurement vector, (10).

End of function MHCForward.

Function to Carry out Reverse Iteration (see Pseudocode 3). The input signals are $\{\widetilde{s}_{\mathrm{A}}\}$ and $\{\widetilde{s}_{\mathrm{B}}\}$. The output is $\{\widetilde{s}_{\mathrm{A}}^{(1)}\}$ which is an enhanced estimate of $\{\widetilde{s}_{\mathrm{A}}\}$ with the contribution from $\{\widetilde{s}_{\mathrm{B}}\}$ removed.

Steps

Step 1. Compute velocity vector, (11).

Step 2. Determine maximum velocity to eliminate velocity vectors that have all components below a certain threshold.

Step 3. Loop over data points and compute the cross-correlation coefficient over a moving data window of P samples.

Step 4. Apply MHC to look for minimum differences between adjacent headings. $\mathbf{d}^-$ and $\mathbf{d}^+$ are used to do this; $\mathbf{d}^+$ which is the sum of adjacent normalized velocity vectors takes into account headings that double back on themselves, corresponding to a peak in the data. If the components of the velocity vectors are larger than a threshold *and* the correlation over P data points is larger than a preset initial value, then indicate this by Flag = 1; otherwise put Flag = 0. If Flag = 1, ΔH is the minimum heading change occurring at $j = j_{\min}$. $\widehat{\mathbf{R}}_{\mathrm{B}}$ is the heading vector at $j = j_{\min}$.

Step 5. If a turning point is found (Flag = 1), then perform the subtraction in (35) using $\widehat{\mathbf{R}}_{\mathrm{B}}$ else do not subtract (Flag = 0). NoSub = Flag is passed to the main routine to indicate whether this subtraction has taken place or not.

End of function MHCReverse.

Acknowledgments

The authors would like to thank and express their appreciation to Professor M. Babaie Zadeh for sending them software for the Clusterwise PCA method [11]. They would also like to thank the four referees for their useful comments.

References

[1] J. J. Rieta, F. Castells, C. Sánchez, V. Zarzoso, and J. Millet, "Atrial activity extraction for atrial fibrillation analysis using blind source separation," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 7, pp. 1176–1186, 2004.

[2] L. Shoker, S. Sanei, and J. Chambers, "Artifact removal from electroencephalograms using a hybrid BSS-SVM algorithm," *IEEE Signal Processing Letters*, vol. 12, no. 10, pp. 721–724, 2005.

[3] Z. Dawy, M. Sarkis, J. Hagenauer, and J. C. Mueller, "A novel gene mapping algorithm based on independent component analysis," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, vol. 5, pp. 381–384, Philadelphia, Pa, USA, March 2005.

[4] D. Nuzillard, S. Bourg, and J. M. Nuzillard, "Model-free analysis of mixtures by NMR using blind source separation," *Journal of Magnetic Resonance*, vol. 133, no. 2, pp. 358–363, 1998.

[5] J. Y. Ren, C. Q. Chang, P. C. W. Fung, J. G. Shen, and F. H. Y. Chan, "Free radical EPR spectroscopy analysis using blind source separation," *Journal of Magnetic Resonance*, vol. 166, no. 1, pp. 82–91, 2004.

[6] W. T. Zhang, S. T. Lou, and Y. L. Zhang, "Robust nonlinear power iteration algorithm for adaptive blind separation of independent signals," *Digital Signal Processing*, vol. 20, no. 2, pp. 541–551, 2010.

[7] M. T. Özgen, E. E. Kuruoğlu, and D. Herranz, "Astrophysical image separation by blind time-frequency source separation methods," *Digital Signal Processing*, vol. 19, no. 2, pp. 360–369, 2009.

[8] Q. He, S. Su, and R. Du, "Separating mixed multi-component signal with an application in mechanical watch movement," *Digital Signal Processing*, vol. 18, no. 6, pp. 1013–1028, 2008.

[9] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Networks*, vol. 13, no. 4-5, pp. 411–430, 2000.

[10] V. Zarzoso and A. K. Nandi, "Blind separation of independent sources for virtually any source probability density function," *IEEE Transactions on Signal Processing*, vol. 47, no. 9, pp. 2419–2432, 1999.

[11] M. Babaie-Zadeh, C. Jutten, and A. Mansour, "Sparse ICA via cluster-wise PCA," *Neurocomputing*, vol. 69, no. 13–15, pp. 1458–1466, 2006.

[12] C. Chang, P. C. W. Fung, and Y. S. Hung, "On a sparse component analysis approach to BSS," in *International Conference on Independent Component Analysis and Blind Source Separation (ICA '06)*, pp. 765–772, Charleston, SC, USA, March 2006.

[13] P. D. O'Grady, B. A. Pearlmutter, and S. T. Rickard, "Survey of sparse and non-sparse methods in source separation," *International Journal of Imaging Systems and Technology*, vol. 15, no. 1, pp. 18–33, 2005.

[14] F. J. Theis, A. Jung, C. G. Puntonet, and E. W. Lang, "Linear geometric ICA: fundamentals and algorithms," *Neural Computation*, vol. 15, no. 2, pp. 419–439, 2003.

[15] P. Georgiev, F. Theis, and A. Cichocki, "Sparse component analysis and blind source separation of underdetermined mixtures," *IEEE Transactions on Neural Networks*, vol. 16, no. 4, pp. 992–996, 2005.

[16] M. Davies and N. Mitianoudis, "Simple mixture model for sparse overcomplete ICA," *IEE Proceedings: Vision, Image and Signal Processing*, vol. 151, no. 1, pp. 35–43, 2004.

[17] M. S. Woolfson, C. Bigan, J. A. Crowe, and B. R. Hayes-Gill, "Method to separate sparse components from signal mixtures," *Digital Signal Processing*, vol. 18, no. 6, pp. 985–1012, 2008.

[18] Y. Sun, C. Ridge, F. Del Rio, A. J. Shaka, and J. Xin, "Postprocessing and sparse blind source separation of positive and partially overlapped data," *Signal Processing*, vol. 91, no. 8, pp. 1838–1851, 2011.

[19] August 2012, http://www.cis.hut.fi/projects/ica/fastica/.

[20] "Database for the Identification of Systems (DaISy)," August 2012, http://homes.esat.kuleuven.be/~smc/daisy/.

[21] Y. Xiang, S. K. Ng, and V. K. Nguyen, "Blind separation of mutually correlated sources using precoders," *IEEE Transactions on Neural Networks*, vol. 21, no. 1, pp. 82–90, 2010.

[22] W. Naanaa and J. M. Nuzillard, "Blind source separation of positive and partially correlated data," *Signal Processing*, vol. 85, no. 9, pp. 1711–1722, 2005.

The Scientific World Journal

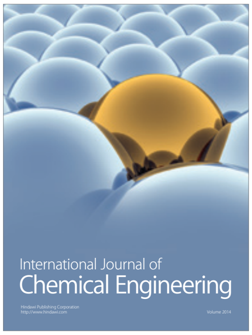
International Journal of
Rotating Machinery

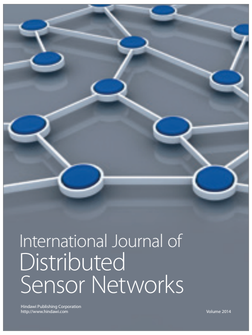
Journal of
Robotics

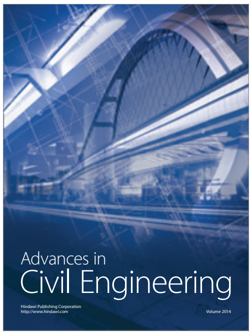
Journal of
Sensors

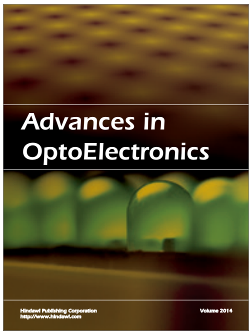
Advances in
Mechanical Engineering

Journal of
Engineering


International Journal of
Chemical Engineering

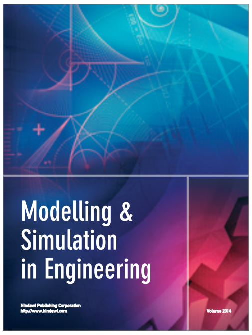

International Journal of
Distributed Sensor Networks

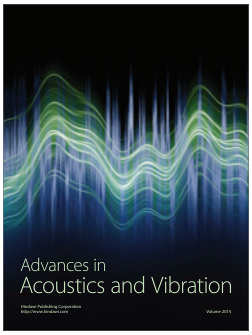

Advances in
Civil Engineering


Advances in
OptoElectronics

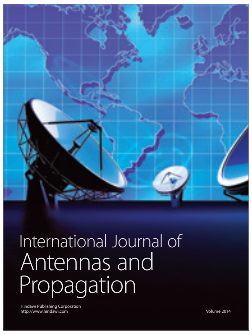
VLSI Design

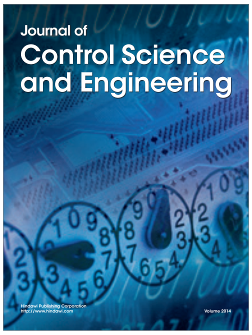
International Journal of
Navigation and Observation

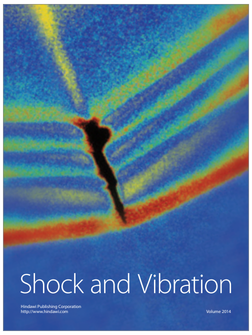

Modelling &
Simulation
in Engineering


Advances in
Acoustics and Vibration

Active and Passive
Electronic Components


International Journal of
Antennas and Propagation


Journal of
Control Science
and Engineering


Shock and Vibration

Journal of
Electrical and Computer
Engineering