# A COMBINED STATISTICS COST FUNCTION FOR BLIND AND SEMI-BLIND SOURCE SEPARATION

*M. Klajman and A.G. Constantinides*

Communications and Signal Processing Research Group
Imperial College of Science, Technology and Medicine
London SW7 2BT

**ABSTRACT**

Most Blind Source Separation algorithms separate the sources by using either second or higher order statistics. In this paper we suggest to use a weighted combination of the second order covariance matrices and the fourth order eigenmatrices to restore the original signals. Thus, we can achieve separation where other algorithms relying on a single type of statistic only, fail. We then develop a novel diagonalisation algorithm, the Cayley Joint Unitary Diagonalisation (CJUD) algorithm, to find the optimal unitary diagonaliser and to determine the weights. Except in some trivial cases, it is very hard to determine the weights without any prior information. We show in this paper how the use of some prior knowledge can be incorporated in the CJUD algorithm in order to get a better estimate of the weights. Simulations are presented to show the improvement in performance of the proposed algorithms.

## 1. INTRODUCTION

Blind Source Separation (BSS) is concerned with recovering the original unknown sources from their observed mixture. The algorithm operates blindly in the sense that, except for statistical independence, no a priori information about either the sources or the transmission medium is known. Most BSS algorithms operate in two steps. First the observed data is whitened. Then the transformation that relates the unknown sources to the decorrelated mixture is found as a pure rotation, since the sources and the decorrelated observations are white vectors. In algebraic BSS methods, this rotation is found by unitarily diagonalising a set of matrices. If the source signals have different spectral contents or are non-stationary, these matrices can be constructed from Second Order Statistics (SOS) only. The SOBI algorithm uses a set of non-zero covariance matrices to separate the sources [1]. If the sources are white, one must resort to Higher Order Statistics (HOS). In JADE, the information required to find the rotation is obtained from the eigenmatrices of the fourth order cumulants [2]. Note however, that HOS based separation is only possible if at most one of the sources is Gaussian. Thus, both SOS and HOS separation algorithms have their specific area of application. In this paper, a slight adjustment to the conventional BSS problem is made: it is assumed that one column of the mixing matrix is known. A similar approach was taken in [3]. In the blind beamforming scenario for example, this would correspond to knowing the array response of one of the sources. First, a new joint diagonalisation algorithm, the Cayley based Joint Unitary Diagonalisation (CJUD) algorithm, is introduced. Then, it is shown how the CJUD algorithm can be changed to incorporate the prior knowledge. Finally, the Combined Weight Statistics (CWS) algorithm is developed. This algorithm separates the sources on the basis of a combination of their SOS and HOS. For a large number of cases, this is possible because most natural signals contain both spectral information and higher order information. The idea of merging different statistics was already suggested in [4]. The contribution of this paper is that the WCS algorithm uses prior information about the mixing matrix to assign different weights to the different statistics. Thus, it can achieve successful separation in difficult cases, e.g. in very noisy circumstances, if the sources are white or if the sources are nearly Gaussian. In all these cases, traditional methods are very likely to fail, either because the signal is drowned in the noise or because there is hardly any additional information available to perform the rotation.

This work was carried out as part of Technology Group TG10 of the MOD Corporate Research Program, supported by the Defence Evaluation Research Agency

## 2. PROBLEM FORMULATION

The instantaneous noiseless BSS problem, with the assumption of an equal number of sources and sensors, can be described mathematically as follows:

$$\mathbf{x}(t) = A\mathbf{s}(t) \quad (1)$$

In this context, the vector $\mathbf{s}(t) = [s_1(t), ..., s_N(t)]^T$ contains the original sources, the vector $\mathbf{x}(t) = [x_1(t), ..., x_N(t)]^T$ contains the array output sampled at time $t$ and $A$ is the $N \times N$ mixing matrix or transfer function between the sources and sensors. In this paper, only real sources will be considered. An implicit assumption is that the sources have unit variance. Separation is achieved if a vector $\mathbf{y}$ can be found so that

$$\mathbf{y}(t) = B\mathbf{x}(t) = PD\mathbf{s}(t) \quad (2)$$

where $B$ is the unmixing matrix, $P$ is a permutation matrix and $D$ is a diagonal matrix. Hence, we can estimate the sources up to their order and their power.

## 3. WHITENING

The whitened observation vector is defined as:

$$\mathbf{z}(t) = W\mathbf{x}(t) \quad (3)$$

where $W$ is the whitening matrix. The covariance matrix of the whitened mixture at lag $\tau$ can then be expressed as:

$$R_z(\tau) = WR_x(\tau)W^T \quad (4)$$

Or in terms of the sources:

$$R_z(\tau) = WAR_s(\tau)A^TW^T \quad (5)$$

A necessary and sufficient condition for the vector $\mathbf{z}(t)$ to be white is that $R_z(0) = I$, where $I$ is the identity matrix. Hence, at zero lag, (5) becomes:

$$R_z(0) = WAA^TW^T = I \quad (6)$$

since the sources have unity power. As $R_x(0) = AA^T$, the whitener can be determined from the observation covariance matrix $R_x(0)$. Equation (6) shows that, provided $W$ is a whitening matrix, $WA$ is a $N \times N$ unitary matrix. It follows that for any whitening matrix $W$, the mixing matrix can be factored as :

$$A = W^{-1}U \quad (7)$$

where $U$ is a $N \times N$ unitary matrix. Finding the whitening matrix $W$ still leaves the unitary factor $U$ of the mixing matrix undetermined. This unknown rotation can be found by exploiting either the time dependence structure of the signals or their fourth order moments.

## 4. FINDING THE ROTATION

### 4.1. Second Order Statistics

The rotation can be found from any covariance matrix of the whitened observations at non-zero lag . Rewriting (5) in terms of the unitary matrix for all non-zero lags, we define:

$$\forall \tau \neq 0 \quad R_z^{\emptyset}(\tau) = UR_s(\tau)U^T \quad (8)$$

As $R_z^{\emptyset}(\tau)$ is a covariance matrix, it is Hermitian and can be diagonalised by a unitary matrix $V$. Moreover, if the eigenvalues of $R_z^{\emptyset}(\tau)$ are distinct, the matrix $V$ will be essentially unique, i.e. up to a scaling and permutation of the columns. Let

$$V^TR_z^{\emptyset}(\tau)V = diag[\lambda_1, ..., \lambda_n] \quad (9)$$

Comparing (8) with (9) and taking into account the essentially uniqueness of $V$, we conclude that $U$ and $V$ are essentially equal, i.e. $V = PDU$. Moreover, the elements $\{\lambda_1, ..., \lambda_n\}$ are a permutation of the autocorrelations of the sources at lag $\tau$. In theory, one covariance matrix at non-zero lag is sufficient to estimate the rotation. In practice, however, it is useful to use a set of matrices as this would enhance the statistical efficiency of the algorithm and prevent an unfortunate choice of lags [1]. From (8), it should be clear that the sources must have some time dependencies. If the sources were white, $R_s(\tau)$ and consequently $R_z^{\emptyset}(\tau)$ would be the null matrix and no additional information would be available to estimate the rotation.

### 4.2. Higher Order Statistics

The cumulant matrices of the whitened observation are defined as:

$$\forall M \quad Q_z(M) : q_{ij} = \sum_{k,l} cum\{z_i, z_j, z_k, z_l\}\, m_{kl} \quad (10)$$

where $m_{kl}$ are the elements of an arbitrary matrix $M$. The cumulant matrix is simply a linear combination of two dimensional slices of a fourth order tensor. Due to the multilinearity and additivity property of the cumulants, (10) can be written as:

$$\forall M \quad Q_z(M) = U\Lambda_MU^T \quad (11)$$

where $\Lambda_M = diag\left\{\kappa_1\mathbf{u}_1^TM\mathbf{u}_1, ..., \kappa_N\mathbf{u}_N^TM\mathbf{u}_N\right\}$, $\kappa_p$ is the kurtosis of the $p$th source and $\mathbf{u}_p$ is the $p$th column of $U$. Equation (11) shows that the rotation matrix can be found as a unitary diagonaliser of the cumulant matrices for any arbitrary matrix $M$. In practice, it is suggested in [2] to use the columns of the unitary matrix $U$ for $M$, i.e. $M = \mathbf{u}_p\mathbf{u}_q^T$ for all $1 \leq p, q \leq N$. Substituting this in (11) will yield the so called eigenmatrices of $\mathbf{z}$:

$$Q_z\left(\mathbf{u}_p\mathbf{u}_q^T\right) = \sum_r \kappa_r \mathbf{u}_r^T\mathbf{u}_p\mathbf{u}_q^T\mathbf{u}_r\mathbf{u}_r\mathbf{u}_r^T \tag{12}$$

For all $p = q$, this becomes:

$$Q_z\left(\mathbf{u}_p\mathbf{u}_p^T\right) = \kappa_p\mathbf{u}_p\mathbf{u}_p^T \tag{13}$$

and zero for all other indices. Jointly diagonalising the different eigenmatrices will yield the unknown unitary factor of the mixing matrix. If the kurtosis of the sources are small, the elements of the eigenmatrices contain too little information to estimate the rotation.

## 5. COMBINED STATISTICS

The cost function of the Weighted Combined Statistics (WCS) algorithm is given by:

$$\begin{aligned}\phi\left(V,\alpha\right) &= \underbrace{\alpha^2\sum_n^N \left\|diag\left[V^TQ_z\left(\mathbf{u}_n\mathbf{u}_n^T\right)V\right]\right\|_F^2}_{HOS} \qquad (14)\\ &\quad + \left(1-\alpha\right)^2\underbrace{\sum_k^K \left\|diag\left[V^TR_z\left(\tau_k\right)V\right]\right\|_F^2}_{SOS}\end{aligned}$$

where $diag\left[A\right]$ denotes the vector formed from the diagonal elements of $A$, $N$ is the number of eigenmatrices, $K$ the number of covariance matrices , $\alpha^2$ and $\left(1-\alpha\right)^2$ are the weights and $\left\|\cdot\right\|_F$ denotes the Frobenius norm. Let

$$R_l\left(\alpha\right) = \begin{cases} \alpha Q_z\left(\mathbf{u}_l\mathbf{u}_l^T\right) & \text{for } 1 \leq l \leq N \\ \left(1-\alpha\right)R_z^{\emptyset}\left(\tau_{l-N}\right) & \text{for } N < l \leq L \end{cases} \tag{15}$$

where $L = K + N$, then (14) can be written as:

$$\phi\left(V,\alpha\right) = \sum_l^L \left\|diag\left[V^TR_l\left(\alpha\right)V\right]\right\|_F^2 \tag{16}$$

For the sake of brevity, the dependency of $R_l$ on $\alpha$ will only be indicated where necessary. Suppose that one column of the mixing matrix is known. In the blind beamforming scenario for example, this would be equivalent to knowing the antenna array response to one signal. The available column will be denoted by $\mathbf{a}$. By using the whitener from (6), one column of the unitary matrix is found as $\mathbf{u} = W\mathbf{a}$. The prior information can then be used in order to formulate the following constraint:

$$\zeta\left(V,\alpha\right) = \left\|\mathbf{a} - \mathbf{v}_p\right\|_F^2 \tag{17}$$

where $\mathbf{v}_p$ is the column of the estimated rotation matrix $V$ corresponding to the given column $\mathbf{u}$. The quantity $\zeta\left(V,\alpha\right)$ simply measures the distance between the actual and estimated array response to the the $p$th source. The semi-BSS problem can then be written as an optimisation problem:

$$\begin{aligned} &\underset{V,\alpha}{\text{maximise}} \quad \phi\left(V,\alpha\right) \\ &\text{subject to} \quad \zeta\left(V,\alpha\right) = 0 \end{aligned} \tag{18}$$

The Lagrangian function is given by:

$$L\left(V,\alpha,\lambda\right) = \underbrace{\sum_l^L \left\|diag\left[V^TR_lV\right]\right\|_F^2}_{\phi(V,\alpha)} + \lambda\underbrace{\left\|\mathbf{u}_p - \mathbf{v}_p\right\|_F^2}_{\zeta(V)} \tag{19}$$

An optimal solution satisfying (18) can be found by using Lagrange Programming Neural Networks (LPNN) [5].

## 6. THE GRADIENTS

### 6.1. The Cayley Transform

The optimal unitary matrix is found by adjusting the previous estimate with a fraction of the gradient, i.e.

$$V\left[k+1\right] = V\left[k\right] + \mu\nabla_V\ L\left(V,\alpha,\lambda\right) \tag{20}$$

where $\mu$ is the adaptation gain and $\nabla_V$ denotes the gradient with respect to $V$. The fundamental problem with (20) is that $V$ will not retain its unitary property. An alternative update equation is required. The Cayley transformation is defined as [6]:

$$C\left(V\right): H = \left(I - V\right)\left(I + V\right)^{-1} \tag{21}$$

and the generalised inverse transform:

$$C^{-1}\left(H\right): V = \left(I - H\right)\left(I + H\right)^{-1} \tag{22}$$

where $V$ and $H$ are respectively a unitary and a skew-Hermitian matrix. Provided that neither $\left(I+V\right)$ nor $\left(I+H\right)$ are singular, (21) transforms a unitary matrix into a skew Hermitian one and (22) reverses the transformation. Let $H$ be a real $N \times N$ skew-Hermitian matrix with elements $h_{ij}$, where $h_{ij} = 0$ if $i = j$ and

$h_{ij} = -h_{ji}$ if $i \neq j$. Define $\partial h_{ij}$ as a small change $\delta$ in element $h_{ij}$ and $-\delta$ in $h_{ji}$. Then:

$$\frac{\partial H}{\partial h_{ij}} = T_{ij} \quad \text{for } 1 \leq j < i \leq N \tag{23}$$

where all the elements of the $N \times N$ matrix $T_{ij}$ are zero except $t_{ij} = 1$ and $t_{ji} = -1$, thus $T_{ij}$ is skew-Hermitian. The gradient of $V$ with respect to $h_{ij}$ is given by:

$$\frac{\partial V}{\partial h_{ij}} = -\left(I + V\right) T_{ij} \left(I + H\right)^{-1} \tag{24}$$

Let $\mathbf{h} = \{h_{ij} : 1 \leq j < i \leq N\}$, then (20) is changed to:

$$H\left[k+1\right] = H\left[k\right] + \mu \nabla_{\mathbf{h}}\, L\left(V, \alpha, \lambda\right) \tag{25}$$

where $\nabla_{\mathbf{h}}\, L\left(V, \alpha, \lambda\right)$ is simply $\frac{\partial L(V,\alpha,\lambda)}{\partial V} \frac{\partial V}{\partial h_{ij}}$ for all $1 \leq j < i \leq N$. As the gradient of $L\left(V, \alpha, \lambda\right)$ will be skew-Hermitian, $H\left[k+1\right]$ will be skew-Hermitian too. At each iteration, $V\left[k\right]$ is transformed into $H\left[k\right]$ using the Cayley transform. Then $H\left[k+1\right]$ is found using (25), finally, $H\left[k+1\right]$ is transformed back into $V\left[k+1\right]$, which will be an updated version of $V\left[k\right]$.

### 6.2. Gradient of the Cost Function

Next, the gradients of the cost function $\phi\left(V, \alpha\right)$ with respect to $\mathbf{h}$ and $\alpha$ are derived. In matrix notation, (16) can be written as:

$$\phi\left(V, \alpha\right) = \sum_{l}^{L} \left\{ diag^T \left[V^T R_l V\right] diag \left[V^T R_l V\right] \right\} \tag{26}$$

The gradient with respect to $\mathbf{h}$ is then given by:

$$\begin{aligned} \frac{\partial \phi\left(V, \alpha\right)}{\partial h_{ij}} &= 2 \sum_{l=1}^{L} diag^T \left[V^T R_l V\right] \\ & \quad diag \left[ V^T \left(R_l + R_l^T\right) \frac{\partial V}{\partial h_{ij}} \right] \end{aligned} \tag{27}$$

where $\frac{\partial V}{\partial h_{ij}}$ is given by (24). Differentiating $\phi\left(V, \alpha\right)$ with respect to $\alpha$ gives:

$$\frac{\partial \phi\left(V, \alpha\right)}{\partial \alpha} = 2 \sum_{l}^{L} \left\{ diag^T \left[V^T R_l V\right] diag \left[ V^T \frac{\partial R_l}{\partial \alpha} V \right] \right\} \tag{28}$$

where:

$$\frac{\partial R_l}{\partial \alpha} = \begin{cases} Q_z\left(\mathbf{u}_l \mathbf{u}_l^T\right) & \text{for } 1 \leq l \leq n \\ -R_z^{\emptyset}\left(\tau_l\right) & \text{for } n < l \leq L \end{cases} \tag{29}$$

A more detailed derivation of (28), (27) and (24) can be found in [7] or obtained from the author in writing.

### 6.3. Gradient of the Constraint

When deriving the gradient of the constraint, it is useful to write (17) in matrix notation:

$$\zeta\left(V, \alpha\right) = \mathbf{e}_p^T \left(F - V\right)^T \left(F - V\right) \mathbf{e}_p \tag{30}$$

where $F$ is the $N \times N$ matrix with the $p$th column of $F$ replaced by $\mathbf{u}$ and $\mathbf{e}_p$ is a $N \times 1$ vector with one at the $p$th position and zero elsewhere. The gradient is then given by:

$$\frac{\partial \zeta\left(V, \alpha\right)}{\partial h_{ij}} = -2 \mathbf{e}_p^H \left(F - V\right)^T \frac{\partial V}{\partial h_{ij}} \mathbf{e}_p \tag{31}$$

The analytical gradient of $\zeta\left(V, \alpha\right)$ with respect to $\alpha$ is very tricky. A first order approximation of the derivative is used instead:

$$\frac{\partial \zeta\left(V, \alpha\right)}{\partial \alpha} \approx \frac{\zeta\left(V, \alpha + \delta\right) - \zeta\left(V, \alpha\right)}{\delta} \tag{32}$$

where $\delta$ indicates a small change.

## 7. THE ALGORITHM

### 7.1. The Generic Form

Using LPNN, the rotation matrix is found by iterating the following equations:

$$\begin{aligned} h_{ij}\left[k+1\right] &= h_{ij}\left[k\right] + \mu \underset{(27)}{\frac{\partial \phi}{\partial h_{ij}}} + \lambda\left[k\right] \underset{(31)}{\frac{\partial \zeta}{\partial h_{ij}}} && (33) \\ \alpha\left[k+1\right] &= \alpha\left[k\right] + \eta \underset{(28)}{\frac{\partial \phi}{\partial \alpha}} + \lambda\left[k\right] \underset{(32)}{\frac{\partial \zeta}{\partial \alpha}} && (34) \\ \lambda\left[k+1\right] &= \lambda\left[k\right] + \rho \underset{(17)}{\zeta\left(V\left[k\right], \alpha\left[k\right]\right)} && (35) \end{aligned}$$

where $\mu$, $\eta$, and $\rho$ are the adaptation gains. For the sake of brevity, the dependencies on $V$ and $\alpha$ have been onmitted, as these are obvious. The gradients in (33) - (35) should be evaluated at $V\left[k\right]$ and $\alpha\left[k\right]$ respectively. At each iteration, (33) must be updated for all the elements of $\mathbf{h}$, i.e. for $h_{ij}$ where $1 \leq j < i \leq N$, so that the matrix $H$ can be constructed and transformed back into the unitary domain. The bracketed numbers under the gradients denote the number of the corresponding equations in the text. Depending on the available information, three different algorithms can be formed from equations (33) - (35).

### 7.2. The CJUD Algorithm

In the ordinary Cayley Joint Unitary Diagonalisation (CJUD) algorithm no a priori information is available.

Moreover, only one type of statistic is used. As $\mathbf{u}$ is unknown, it is replaced by any column $\mathbf{v}_p$ of the estimated rotation $V$ so that $\zeta(V, \alpha) = 0$. Hence, (35) and the last term of (33) can be ignored. For HOS, $\alpha$ is set to one, so that the matrices $R_l$ in (15) will consist of eigenmatrices only. Alternatively, $\alpha = 0$ if only SOS are used. As $\alpha$ is constant, (34) can be ignored as well so that the only required update equation is $\mathbf{h}[k+1] = \mathbf{h}[k] + \mu \frac{\partial \phi}{\partial \mathbf{h}}$.

### 7.3. The CCJUD Algorithm

In the Constrained Cayley Joint Unitary Diagonalisation (CCJUD) algorithm, some prior information is given in the form of a column of the mixing matrix. As with the CJUD, the value of $\alpha$ is fixed according to the use of the statistics and (34) can again be ignored. The rotation is found by using (33) and (35).

### 7.4. The WCS Algorithm

The Weighted Combined Statistics (WCS) algorithm uses a combination of higher and second order statistics to estimate the mixing matrix. The weights assigned to the different statistics are determined by the prior information. The matrices $R_l$ are formed using (15). Cycling through (33), (34) and (35) will yield an estimation of the rotation. Due to space limitations, it is impossible to discuss the computational aspects extensively, but two comments are in order here. Experience has shown that for best results, it is advisable to set $\eta$ in (34) to zero. Thus, the selection of the weight is only determined by the constraint and not by the cost function. Secondly, $\frac{\partial \zeta}{\alpha}$ can be estimated efficiently using the CJUD algorithm, initialising it with $V[k]$.

## 8. SIMULATIONS

In the first experiment, a comparision between the CCJUD algorithm using prior knowledge, and the JADE algorithm is done. Three artificial signals of sample length 1000 are created. The first two signals are a sine wave and a random Gaussian sequence. The sequence $a(n)$ is a discrete i.i.d signal called $MS(\beta)$ that takes its values from the set $\{-1, 0, \beta\}$ with the respective probabilities $\{1/(1+\beta), (\beta-1)/\beta, 1/(\beta(1+\beta))\}$ . The so called cumulant parameter $\beta$ is chosen so that $\beta \geq 1$. It is easily verified that $E\{a\} = 0$, $E\{a^2\} = 1$ and $cum\{a, a, a, a\} = \beta^2 - \beta - 2$. The third signal was chosen as $MS(2.1)$. Thus, the last two signals are approximately non-kurtic and separation on the basis of HOS will prove to be difficult. The signals were mixed with a $3 \times 3$ mixing matrix with elements picked from a Gaussian distribution. No noise was added.

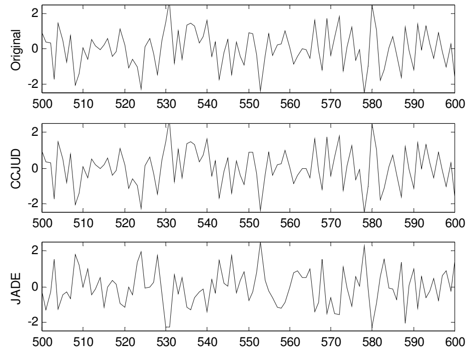


Figure 1: The original second signal (top), its reconstruction by CCJUD (middle) and by JADE (bottom)

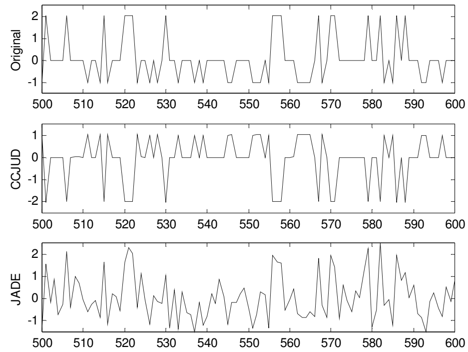


Figure 2: The original third signal (top), its reconstruction by CCJUD (middle) and by JADE (bottom)

Due to lack of space, the separation of the sine wave is not shown. It was successfully reconstructed by both algorithms. The CCJUD algorithm achieves virtually perfect separation. Note that the separated version of the $MS(2.1)$ signal, middle figure in fig.2 has to be multiplied by minus one. The JADE algorithm however, is unable to separate the Gaussian signal from the $MS(2.1)$ signal. This is hardly suprising, the second signal is Gaussian and the third signal has a very small fourth order cumulant so there is not enough information to estimate the rotation. The $PI(P)$ index measures the distance between $P$, which is the inverse of the estimated mixing matrix multiplied by the true mixing matrix, and a permutation matrix [7]. The smaller the $PI(P)$, the better the separation. The $PI(P)$ measure for JADE and CCJUD are 2.25 and 0.05 respectively, hence the CCJUD algorithm performs significantly better. It is unfair to compare the performance of JADE with CCJUD, as the latter uses some prior knowledge. However, the aim of these graphs is to show that some prior knowledge can enhance performance significantly. Consider a situation where $n$ sources are transmitting simulataneously. If separation is impossible, $n-1$ sources can keep quiet for a while so that the array response to one source can be estimated. Using the CCJUD algorithm, it is then possible to achieve complete separation. In the second experiment, the performance of the Weighted Combined Statistics (WSCS) algorithm is evaluated. Three algorithms are compared. The $\mathrm{JADE_{TD}}$ algorithm [4] operates completely blindly and assigns equal weights to the HOS and SOS. Similarly, the CCJUD algorithm uses equal weights for both statistics. However, unlike the $\mathrm{JADE_{TD}}$, it uses some prior information. Finally, the WCS algorithm uses the prior information to assign different weights to the SOS and the HOS. The three sources used are $MS(2.2)$, $MS(2.3)$ and $MS(2.4)$ passed through a IIR(2) filter with complex conjugate poles at $\exp(\pm.7\pi j)$, $\exp(\pm.6\pi j)$ and $\exp(\pm.6\pi j)$. Since the sources are nearly non-kurtic and their spectral content is very similar, separation based on either HOS or SOS alone, will not succeed. As in the previous experiment, a $3 \times 3$ mixing matrix is used with elements from a real Gaussian distribution. The performance measure used is the Interference to Signal Ratio (ISR) index, $ISR_l(P) = \sum_{k \neq l} \frac{|p_{kl}|}{|p_l|}$, where $p_{kl}$ are the elements of $P$. The $ISR(P)$ simply measures the amount of interference from the undesired signals that is present in the desired signal. The smaller the $ISR(P)$, the better the separation. The experiment was averaged over 20 trials. A detailled discussion of fig.3 is beyond the scope of this paper. Note however the significant improvent of the WCS algorithm on the CCJUD algorithm despite the fact that both use the same amount of information. The $PI(P)$ for $\mathrm{JADE_{TD}}$, CCJUD and WCS are 0.90, 0.78 and 0.63 respectively.

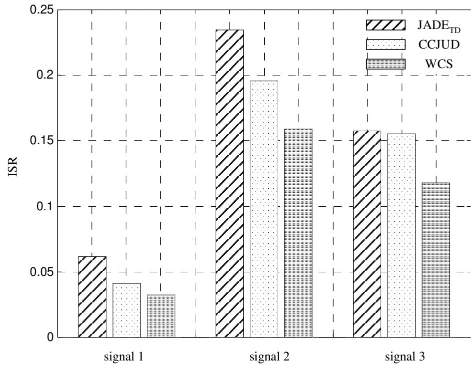


Figure 3: The Interference to Signal Ratio of the three signal for $\mathrm{JADE_{TD}}$ (striped), CCJUD (dotted) and WCS (solid).

## 9. CONCLUSION

Three novel algorithms were presented in this paper. We first introduced an algorithm that finds a joint unitary diagonaliser of a given set of matrices. It was then shown how this can be applied to the BBS problem. A simple modification to the CJUD algorithm permits it to consider some prior information. Finally, the CWS was presented, which makes use of the prior knowledge in order to assign the weights to the statistics.

## REFERENCES

[1] A. Belouchrani, K. Abed-Meraim, J. Cardoso, and E. Moulines, "A blind source separation technique using second order statistics," *IEEE Transactions on Signal Processing*, vol. 45, no. 2, pp. 434–444, 1997.

[2] J. F. Cardoso and A. Soulomiac, "Blind beamforming for non-gaussian signals," *IEE PROCEEDINGS-F*, vol. 140, pp. 362 – 370, December 1993.

[3] J. Igual and L. Vergara, "Prior information about mixing matrix in BSS-ICA formulation," in *ICA-2000*, pp. 123 – 126, 2000.

[4] P. P. K. R. Robert Muller and A. Ziehe, "JADE TD; combining higher order statistics and temporal information for blind source separation (with noise)," in *ICA-99*, pp. 87 – 92, 1999.

[5] S. Zhang and A. G. Constantinides, "Lagrange programming neural networks," *IEEE Transactions on Circuits and Systems II: Analog Digital Signal Processing*, vol. 39, pp. 441– 452, July 1992.

[6] R. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge University Press, 1999.

[7] M. Klajman and J. A. Chambers, "Approximate joint diagonalisation based on the cayley transformation," *Proceedings of the IMA, Oxford Press*, 2000.