WEIGHTED ACMA

Alle-Jan van der Veen

Delft University of Technology, Dept. Electrical Engineering/DIMES, 2628 CD Delft, The Netherlands

The analytical constant modulus algorithm (ACMA) is a deterministic array processing algorithm to separate sources based on their constant modulus. It has been derived without detailed regard to noise processing. In particular, the estimates of the beamformer are known to be asymptotically biased. In the present paper, we investigate this bias, and obtain a straightforward weighting scheme that will whiten the noise and remove the bias. This leads to improved performance for larger data sets.

1. INTRODUCTION

Constant modulus algorithms (CMAs) enjoy widespread popularity as methods for blind source separation and equalization. The original CMA [1, 2] was developed for the purpose of equalization and is an LMS-type iteration. Other algorithms are block-iterative. There are numerous modifications and enhancements, especially with regard to initialization and convergence issues.

One aspect that distinguishes source separation from equalization is that it is desired to recover *all* impinging CM signals. Successive cancellation algorithms (in which one signal is retrieved and removed using LMS) have be defined but are sensitive and need long convergence times [3], parallel cancellation algorithms need good initializations or an additional 'independence condition' in order to converge to different signals [4].

An algorithm that solves the instantaneous CM separation problem elegantly in a non-iterative algebraic way is the recently derived 'Analytic CMA' (ACMA) [5]. The problem is formulated as an overdetermined system of quadratic equations, whose solution can be found by solving a linear system followed by a generalized eigenvalue problem. This algorithm is quite robust, even on very small data sets, and shows good results on measured data [6]. However, its performance has not been analyzed yet.

This paper makes a start at such an analysis by investigating the noise contribution at the first step, the solution of the linear system. Since the entries of the corresponding matrix are essentially cross-multiplications of the data samples, it is seen that the noise on this matrix is not white, leading to a bias in the estimated solution and a suboptimal asymptotic performance. The main contribution of the paper is the derivation of an expression for this bias. Subsequently, the algorithm is extended by a noise whitening step which almost removes the bias. It is demonstrated that this greatly improves the performance for large data sets with closely spaced sources.

Notation Vectors are denoted by boldface, matrices by capitals. Overbar $\overline{(\ )}$ denotes complex conjugation, ^{T} is the matrix transpose, * the matrix complex conjugate transpose. I_m is the $m\times m$ identity matrix, $\mathbf{0}$ and $\mathbf{1}$ are vectors for which all entries are equal to 0 and 1, respectively. $\otimes$ is the Kronecker product, $\circ$ is the "Khatri-Rao" product, which is a column-wise Kronecker product:

$$A\otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots \\ a_{21}B & a_{22}B & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}, \quad A\circ B = [\mathbf{a}_1\otimes\mathbf{b}_1 \quad \mathbf{a}_2\otimes\mathbf{b}_2 \quad \cdots].$$

Two notable properties are: $\mathrm{vec}(ABC) = (C^T\otimes A)\mathrm{vec}(B)$, and $\mathrm{vec}(\mathbf{a}\mathbf{b}^*) = \bar{\mathbf{b}}\otimes\mathbf{a}$, where the vec-operator indicates a stacking of the columns of a matrix into a vector.

Finally, E denotes the expectation operator, and $\sim$ indicates equality in expectation.

2. DATA MODEL

Consider d independent sources, transmitting signals $s_i(t)$ with constant modulus waveforms ($|s_i(t)| = 1$) in a wireless scenario. The signals are received by an array of M antennas. We stack the antenna outputs $x_i(t)$ into vectors $\mathbf{x}(t)$ and collect N samples in a matrix $X : M\times N$. Assuming that the sources are sufficiently narrowband in comparison to the delay spread of the multipath channel, this leads to the well-known data model

$$X = AS = \mathbf{a}_1\mathbf{s}_1 + \cdots + \mathbf{a}_d\mathbf{s}_d\,. \tag{1}$$

$A = [\mathbf{a}_1 \ \cdots \mathbf{a}_d] \in \mathbb{C}^{M\times d}$ is the array response matrix with columns $\mathbf{a}_i$. The rows $\mathbf{s}_i$ of $S\in\mathbb{C}^{d\times N}$ contain the samples of the source signals.

In the blind signal separation scenario, both A and S are unknown, and the objective is, given X, to find the factorization $X = AS$. Alternatively, we try to find a beamforming matrix $W\in\mathbb{C}^{d\times M}$ of full row rank d such that $S = WX$. Note that for source separation using beamforming to be possible, we need $d\le M$ and A full rank, so that it has a left inverse W.

Constant-modulus algorithms try to find the factorization $X = AS$ based on the constant-modulus property of S, i.e.,

$$|S_{ik}| = 1\,.$$

With additive noise, the data model is

$$\tilde{X} = AS + E\,.$$

3. ORIGINAL ACMA ALGORITHM

We consider first the basic ACMA algorithm for the noiseless case. The objective is to find all beamforming vectors $\mathbf{w}$ that reconstruct a signal with a constant modulus, i.e.,

$$\mathbf{w}^*X = \mathbf{s}\,, \qquad |s_k| = 1 \quad (k = 1,\cdots,N)\,. \tag{2}$$

It is known that, if sufficiently many samples are taken, $\mathbf{s}$ will be one of the original source signals. Let $\mathbf{x}_k$ be the k-th column of X. By substitution, we find

$$\mathbf{w}^*\mathbf{x}_k\mathbf{x}_k^*\mathbf{w} = 1 \quad\Leftrightarrow\quad (\bar{\mathbf{x}}_k\otimes\mathbf{x}_k)^*(\bar{\mathbf{w}}\otimes\mathbf{w}) = 1 \quad (k = 1,\cdots,N)\,.$$

Thus define

$$P = [\bar{X}\circ X]^* = \begin{bmatrix} \mathbf{p}_1^* \\ \vdots \\ \mathbf{p}_N^* \end{bmatrix}, \qquad \mathbf{p}_k = \bar{\mathbf{x}}_k\otimes\mathbf{x}_k\,.$$

Then (2) is equivalent to finding all $\mathbf{w}$ that satisfy

$$P\mathbf{y} = \mathbf{1}\,, \qquad \mathbf{y} = \bar{\mathbf{w}} \otimes \mathbf{w}\,.$$

The ACMA technique solves this problem by the following three steps:

1. Find a basis $\{\mathbf{y}_1, \cdots, \mathbf{y}_\delta\}$ of all solutions of the linear system

   $$P\mathbf{y} = \alpha \mathbf{1}\,, \qquad \alpha \in \mathbb{C}$$

   where α is arbitrary.

2. Find all linear combinations of the basis vectors that have the required structure

   $$\bar{\mathbf{w}} \otimes \mathbf{w} = \alpha_1 \mathbf{y}_1 + \cdots + \alpha_\delta \mathbf{y}_\delta\,, \qquad \alpha_i \in \mathbb{C}\;.$$

   This gives d independent solutions $\mathbf{w}_1, \cdots, \mathbf{w}_d$.

3. Scale each solution $\mathbf{w}_i$ such that the average output power

   $$\frac{1}{N}\sum_{k=1}^{N} \mathbf{w}_i^* \mathbf{x}_k \mathbf{x}_k^* \mathbf{w}_i = \mathbf{w}_i^* \left(\sum_{k=1}^{N} \mathbf{x}_k \mathbf{x}_k^* \right) \mathbf{w}_i \qquad (3)$$

   is equal to 1. This ensures that $\alpha = 1$.

The second step is shown to be equivalent to a generalized eigenvalue problem, provided that $\delta = d$. (For this it is necessary that X has full rank, and a preprocessing is needed. See section 5.) The first step requires the solution of an overdetermined linear system of equations. This problem can be cast into more practical formulations, as follows.

Let Q be any unitary matrix such that

$$Q[\mathbf{1} \quad P] =: \begin{bmatrix} \sqrt{N} & \mathbf{p}^* \\ \mathbf{0} & P' \end{bmatrix},$$

i.e., Q zeroes the entries of the first column. Q can e.g., be computed from a QR factorization of $[\mathbf{1} \quad P]$. Then

$$P\mathbf{y} = \alpha\mathbf{1} \quad \Leftrightarrow \quad Q[\mathbf{1} \quad P] \begin{bmatrix} -\alpha \\ \mathbf{y} \end{bmatrix} = \mathbf{0} \quad \Leftrightarrow \quad \begin{cases} \mathbf{p}^*\mathbf{y} = \sqrt{N}\alpha \\ P'\mathbf{y} = \mathbf{0} \end{cases}$$

The second equation says that $\{\mathbf{y}_i\}$ is a basis for the null space of the matrix P', and it can be conveniently found from an SVD of P'. Since α is free, the first equation is of no importance (as shown below, it is equal to the condition in step 3).

Define $R = P'^* P'$. Instead of analyzing the influence of noise on P', it will be more convenient to analyze R.

Lemma 1. *R satisfies*

$$\begin{aligned} R &= P^*P - \tfrac{1}{N} P^* \mathbf{1}\mathbf{1}^* P \\ &= \textstyle\sum (\bar{\mathbf{x}}_k \otimes \mathbf{x}_k)(\bar{\mathbf{x}}_k \otimes \mathbf{x}_k)^* - \tfrac{1}{N} \left[\sum \bar{\mathbf{x}}_k \otimes \mathbf{x}_k\right] \left[\sum \bar{\mathbf{x}}_k \otimes \mathbf{x}_k\right]^*\,. \end{aligned} \qquad (4)$$

Moreover,

$$P\mathbf{y} = \alpha\mathbf{1} \;\; (\alpha \in \mathbb{C}\,) \quad \Leftrightarrow \quad \mathbf{y}^* R \mathbf{y} = 0\,.$$

PROOF Partition Q as

$$Q = \begin{bmatrix} \mathbf{q}_1^* \\ Q' \end{bmatrix}$$

From $Q'\mathbf{1} = \mathbf{0}$ and the unitarity of Q, it follows that $\mathbf{q}_1 = \frac{1}{\sqrt{N}}\mathbf{1}$. Thus

$$Q'^* Q' = Q^* Q - \mathbf{q}_1 \mathbf{q}_1^* = I - \frac{1}{N}\mathbf{1}\mathbf{1}^*$$

With $P' = Q'P$ it follows that $R = P'^*P' = P^*(I - \frac{1}{N}\mathbf{1}\mathbf{1}^*)P = P^*P - \frac{1}{N}P^*\mathbf{1}\mathbf{1}^*P$.

To prove the second part, note that '$\Rightarrow$' follows immediately by substitution into (4). For '$\Leftarrow$', we could make an argument using $R = P'^*P'$, but instead we will give a more general proof using only (4):

$$\begin{aligned} & \mathbf{y}^* R \mathbf{y} = 0 \\ \Leftrightarrow \quad & \det \begin{bmatrix} \mathbf{1}^*\mathbf{1} & \mathbf{1}^* P \mathbf{y} \\ \mathbf{y}^* P^* \mathbf{1} & \mathbf{y}^* P^* P \mathbf{y} \end{bmatrix} = 0 \\ \Leftrightarrow \quad & \exists \alpha : \quad [-\alpha^* \quad 1] \begin{bmatrix} \mathbf{1}^*\mathbf{1} & \mathbf{1}^* P \mathbf{y} \\ \mathbf{y}^* P^* \mathbf{1} & \mathbf{y}^* P^* P \mathbf{y} \end{bmatrix} \begin{bmatrix} -\alpha \\ 1 \end{bmatrix} = 0 \\ \Leftrightarrow \quad & \exists \alpha : \quad [-\alpha^* \quad \mathbf{y}^*] \begin{bmatrix} \mathbf{1}^* \\ P^* \end{bmatrix} [\mathbf{1} \quad P] \begin{bmatrix} -\alpha \\ \mathbf{y} \end{bmatrix} = 0 \\ \Leftrightarrow \quad & \exists \alpha : \quad P\mathbf{y} = \alpha \mathbf{1}\,. \end{aligned}$$

□

Incidentally, note that $\mathbf{p}^* = \frac{1}{\sqrt{N}}\mathbf{1}^*P = \frac{1}{\sqrt{N}}\sum(\bar{\mathbf{x}}_k \otimes \mathbf{x}_k)^*$, so that the first equation $\mathbf{p}^*\mathbf{y} = \sqrt{N}\alpha$ defines α as $\alpha = \frac{1}{N}\sum(\bar{\mathbf{x}}_k \otimes \mathbf{x}_k)^*\mathbf{y} = \frac{1}{N}\sum \mathbf{w}^*\mathbf{x}_k\mathbf{x}_k^*\mathbf{w}$. Thus, α is interpreted as the average output power of the beamformer, and it is set to 1 in the 3rd step.

4. EFFECT OF NOISE

With the preceding Lemma, we can describe the basis $\{\mathbf{y}_i\}$ to be computed in the first step of ACMA as a basis of the null space of P', i.e., a basis of the null space of R.

Let us now assume that our observations are noise perturbed:

$$\tilde{\mathbf{x}}_k = \mathbf{x}_k + \mathbf{e}_k\,, \qquad k = 1, \cdots, N\,,$$

and that we compute in the same way as before

$$\tilde{R} = \tilde{P}^*\tilde{P} - \frac{1}{N}\tilde{P}^*\mathbf{1}\mathbf{1}^*\tilde{P}\,, \qquad (5)$$

where now $\tilde{P} = [\bar{\tilde{X}} \circ \tilde{X}]^*$. We analyze the contribution of the noise in this expression. (For readability, we will drop the subscript k in $\mathbf{x}$ and $\mathbf{e}$ if it does not lead to confusion.)

We assume zero mean, circularly symmetric noise independent of the sources, and define

$$\mathrm{E}(\mathbf{e}\mathbf{e}^*) =: \sigma^2 R_e\,, \qquad \mathrm{E}[(\bar{\mathbf{e}} \otimes \mathbf{e})(\bar{\mathbf{e}} \otimes \mathbf{e})^*] =: \sigma^4 C_e\,.$$

For i.i.d. white gaussian noise with variance σ^2, we have $R_e = I$ and $C_e = I + \mathrm{vec}(I)\mathrm{vec}(I)^T$.

Theorem 2. *Define $\tilde{R}$ as in (5). With the above assumptions on the noise,*

$$\tilde{R} \sim R + \sigma^2 R_n + \sigma^4 C_n$$

where

$$\begin{aligned} R &= \textstyle\sum(\bar{\mathbf{x}} \otimes \mathbf{x})(\bar{\mathbf{x}} \otimes \mathbf{x})^* - \frac{1}{N}\left[\sum \bar{\mathbf{x}} \otimes \mathbf{x}\right]\left[\sum \bar{\mathbf{x}} \otimes \mathbf{x}\right]^* \\ R_n &= (\textstyle\sum \mathbf{x}\mathbf{x}^*)^T \otimes R_e + R_e^T \otimes (\sum \mathbf{x}\mathbf{x}^*) \\ C_n &= NC_e - N\mathrm{vec}(R_e)\mathrm{vec}(R_e)^*\,. \end{aligned}$$

PROOF The first term in the definition of $\tilde{R}$ gives

$$\begin{aligned} \tilde{P}^*\tilde{P} &= [\bar{\tilde{X}} \circ \tilde{X}][\bar{\tilde{X}} \circ \tilde{X}]^* \\ &= \textstyle\sum_1^N (\bar{\mathbf{x}} \otimes \mathbf{x} + \bar{\mathbf{x}} \otimes \mathbf{e} + \bar{\mathbf{e}} \otimes \mathbf{x} + \bar{\mathbf{e}} \otimes \mathbf{e}) \cdot \\ &\quad \cdot (\bar{\mathbf{x}} \otimes \mathbf{x} + \bar{\mathbf{x}} \otimes \mathbf{e} + \bar{\mathbf{e}} \otimes \mathbf{x} + \bar{\mathbf{e}} \otimes \mathbf{e})^* \\ &= \textstyle\sum_1^N \Big[(\bar{\mathbf{x}} \otimes \mathbf{x})(\bar{\mathbf{x}} \otimes \mathbf{x})^* + (\bar{\mathbf{x}} \otimes \mathbf{e})(\bar{\mathbf{x}} \otimes \mathbf{e})^* \\ &\qquad + (\bar{\mathbf{e}} \otimes \mathbf{x})(\bar{\mathbf{e}} \otimes \mathbf{x})^* + (\bar{\mathbf{e}} \otimes \mathbf{e})(\bar{\mathbf{e}} \otimes \mathbf{e})^* \\ &\qquad + (\bar{\mathbf{x}} \otimes \mathbf{x})(\bar{\mathbf{e}} \otimes \mathbf{e})^* + (\bar{\mathbf{e}} \otimes \mathbf{e})(\bar{\mathbf{x}} \otimes \mathbf{x})^* \\ &\qquad + ① + ② + ③ \Big] \end{aligned}$$

where

$$\begin{aligned} ① &= (\bar{\mathbf{x}}\otimes\mathbf{x})(\bar{\mathbf{x}}\otimes\mathbf{e})^* + (\bar{\mathbf{x}}\otimes\mathbf{x})(\bar{\mathbf{e}}\otimes\mathbf{x})^* \\ &\quad + (\bar{\mathbf{x}}\otimes\mathbf{e})(\bar{\mathbf{x}}\otimes\mathbf{x})^* + (\bar{\mathbf{e}}\otimes\mathbf{x})(\bar{\mathbf{x}}\otimes\mathbf{x})^* \\ ② &= (\bar{\mathbf{x}}\otimes\mathbf{e})(\bar{\mathbf{e}}\otimes\mathbf{x})^* + (\bar{\mathbf{e}}\otimes\mathbf{x})(\bar{\mathbf{x}}\otimes\mathbf{e})^* \\ ③ &= (\bar{\mathbf{x}}\otimes\mathbf{e})(\bar{\mathbf{e}}\otimes\mathbf{e})^* + (\bar{\mathbf{e}}\otimes\mathbf{x})(\bar{\mathbf{e}}\otimes\mathbf{e})^* \\ &\quad + (\bar{\mathbf{e}}\otimes\mathbf{e})(\bar{\mathbf{x}}\otimes\mathbf{e})^* + (\bar{\mathbf{e}}\otimes\mathbf{e})(\bar{\mathbf{e}}\otimes\mathbf{x})^* . \end{aligned}$$

The assumptions on the noise imply

$$\mathrm{E}(\mathbf{e}) = 0, \quad \mathrm{E}(\mathbf{e}\mathbf{e}^T) = 0, \quad \mathrm{E}(\mathbf{e}\mathbf{e}^*\mathbf{e}) = 0$$

so that ① ~ 0, ② ~ 0, ③ ~ 0.

Use the relations $(\mathbf{a}\otimes\mathbf{b})(\mathbf{c}\otimes\mathbf{d})^* = \mathbf{a}\mathbf{c}^*\otimes\mathbf{b}\mathbf{d}^*$ and $\mathrm{vec}(\mathbf{a}\mathbf{b}^*) = \bar{\mathbf{b}}\otimes\mathbf{a}$ to obtain

$$\begin{aligned} (\bar{\mathbf{x}}\otimes\mathbf{e})(\bar{\mathbf{x}}\otimes\mathbf{e})^* &= \bar{\mathbf{x}}\mathbf{x}^T\otimes\mathbf{e}\mathbf{e}^* \sim \bar{\mathbf{x}}\mathbf{x}^T\otimes\sigma^2 R_e \\ (\bar{\mathbf{e}}\otimes\mathbf{x})(\bar{\mathbf{e}}\otimes\mathbf{x})^* &= \bar{\mathbf{e}}\mathbf{e}^T\otimes\mathbf{x}\mathbf{x}^* \sim \sigma^2 R_e^T\otimes\mathbf{x}\mathbf{x}^* \\ (\bar{\mathbf{x}}\otimes\mathbf{x})(\bar{\mathbf{e}}\otimes\mathbf{e})^* &\sim \sigma^2(\bar{\mathbf{x}}\otimes\mathbf{x})\mathrm{vec}(R_e)^* \\ (\bar{\mathbf{e}}\otimes\mathbf{e})(\bar{\mathbf{x}}\otimes\mathbf{x})^* &\sim \sigma^2\mathrm{vec}(R_e)(\bar{\mathbf{x}}\otimes\mathbf{x})^* \\ (\bar{\mathbf{e}}\otimes\mathbf{e})(\bar{\mathbf{e}}\otimes\mathbf{e})^* &\sim \sigma^4 C_e . \end{aligned}$$

These are inserted in the expression for $\tilde{P}^*\tilde{P}$. Similarly, we find for the second term

$$\begin{aligned} \tilde{P}^*\mathbf{1} &= \sum[\bar{\mathbf{x}}\otimes\mathbf{x} + \bar{\mathbf{x}}\otimes\mathbf{e} + \bar{\mathbf{e}}\otimes\mathbf{x} + \bar{\mathbf{e}}\otimes\mathbf{e}] \\ &\sim \sum(\bar{\mathbf{x}}\otimes\mathbf{x}) + 0 + 0 + \sigma^2 N\mathrm{vec}(R_e) \end{aligned}$$

and

$$\begin{aligned} \tilde{P}^*\mathbf{1}\mathbf{1}^*\tilde{P} \sim\ & [\textstyle\sum\bar{\mathbf{x}}\otimes\mathbf{x}]\,[\sum\bar{\mathbf{x}}\otimes\mathbf{x}]^* \\ & +\sigma^2 N[\textstyle\sum\bar{\mathbf{x}}\otimes\mathbf{x}]\,\mathrm{vec}(R_e)^* + \sigma^2 N\mathrm{vec}(R_e)\,[\sum\bar{\mathbf{x}}\otimes\mathbf{x}]^* \\ & +\sigma^4 N^2\mathrm{vec}(R_e)\mathrm{vec}(R_e)^* . \end{aligned}$$

Piecing everything together in the expression for $\tilde{R}$, a number of terms cancel, and we obtain the claimed result. □

Thus $\tilde{R}$ is in expectation equal to the noise-free R, a second-order contribution R_n due to noise, and a fourth-order contribution which is hopefully insignificant.

For white gaussian noise,

$$\begin{aligned} \tilde{R} \sim\ & \textstyle\sum(\bar{\mathbf{x}}\otimes\mathbf{x})(\bar{\mathbf{x}}\otimes\mathbf{x})^* - \frac{1}{N}[\sum\bar{\mathbf{x}}\otimes\mathbf{x}]\,[\sum\bar{\mathbf{x}}\otimes\mathbf{x}]^* \\ & + \sigma^2\left[\textstyle\sum\bar{\mathbf{x}}\mathbf{x}^T\otimes I + I\otimes\sum\mathbf{x}\mathbf{x}^*\right] + N\sigma^4 I . \end{aligned}$$

If we assume that $\|\sigma^2 I_M\|_F \ll \|\frac{1}{N}\sum\mathbf{x}\mathbf{x}^*\|_F$, i.e., a sufficiently large SNR, then we can ignore the fourth-order term. Note that the relative size of this approximation is independent of N: as a result, the estimates will have a small asymptotic bias.

Noise whitening

Let us assume that we know the noise covariance up to a scalar, i.e., we know R_e. We cannot know R_n since it depends on noise-free data, but we can construct

$$\tilde{R}_n := (\textstyle\sum\bar{\tilde{\mathbf{x}}}\tilde{\mathbf{x}}^T)\otimes R_e + R_e^T\otimes(\sum\tilde{\mathbf{x}}\tilde{\mathbf{x}}^*) .$$

It is straightforward to show (with $\tilde{\mathbf{x}} = \mathbf{x}+\mathbf{e}$) that

$$\tilde{R}_n \sim R_n + \sigma^2 N[R_e^T\otimes R_e + R_e^T\otimes R_e] .$$

If we again assume that $\|\sigma^2 R_e\|_F \ll \|\frac{1}{N}\sum\mathbf{x}\mathbf{x}^*\|_F$, i.e., a sufficiently large SNR, then the correction by the second term is small, so that $\tilde{R}_n \simeq R_n$.

Thus, we have available the data matrices $\tilde{R}$ and $\tilde{R}_n$, satisfying the model (ignoring 4-th order terms)

$$\tilde{R} \simeq R + \sigma^2\tilde{R}_n .$$

Since R is rank deficient with a kernel of dimension δ, we can estimate σ^2 as the (average of the) smallest δ eigenvalues of the matrix pencil $(\tilde{R}, \tilde{R}_n)$, corresponding to the eigenvalue equation

$$(\tilde{R} - \lambda\tilde{R}_n)\mathbf{y} = 0 .$$

An estimate of the basis $\{\mathbf{y}_i\}$ of the kernel of R is given by the corresponding eigenvectors.

Alternatively, we can use $\tilde{R}_n^{1/2}$ to prewhiten the data. Recall the factorization $\tilde{R} = \tilde{P}'^*\tilde{P}'$, where in fact $\tilde{P}'$ is obtained from a QR factorization of $[\mathbf{1}\ \tilde{P}]$.

$$\begin{aligned} & (\tilde{R} - \lambda\tilde{R}_n)\mathbf{y} = 0 \\ \Leftrightarrow\ & \tilde{R}_n^{1/2}(\tilde{R}_n^{-1/2}\tilde{R}\tilde{R}_n^{-1/2} - \lambda I)\tilde{R}_n^{1/2}\mathbf{y} = 0 \\ \Leftrightarrow\ & (\tilde{P}''^*\tilde{P}'' - \lambda I)\mathbf{y}' = 0 , \end{aligned}$$

where

$$\begin{aligned} \tilde{P}'' &:= \tilde{P}'\tilde{R}_n^{-1/2} \\ \mathbf{y}' &:= \tilde{R}_n^{1/2}\mathbf{y}_i . \end{aligned}$$

Thus we compute $\{\mathbf{y}_i'\}$ as the δ least significant right singular vectors of $\tilde{P}'\tilde{R}_n^{-1/2}$, and then set $\mathbf{y}_i = \tilde{R}_n^{-1/2}\mathbf{y}_i'$.

5. DETAILS

Prewhitening and rank truncation

Suppose we premultiply $\tilde{X}$ with any invertible matrix F. Then R is replaced by $(\bar{F}\otimes F)R(\bar{F}\otimes F)^*$ and R_n by $(\bar{F}\otimes F)R_n(\bar{F}\otimes F)^*$. Thus the basis of the null space will be transformed by $(\bar{F}\otimes F)^{-*}$ but obviously, this has no effect on the resulting beamformers. Hence, a prewhitening of the data matrix to reduce R_e to I is not essential.

However, a preliminary transformation is useful for the following reason. The rank of X is d, thus if $d < M$, then X is rank deficient. X has to be full rank or else the null space of P is inflated with $M^2 - d^2$ additional vectors $\mathbf{y}$. These satisfy $P\mathbf{y} = \alpha\mathbf{1}$ (with $\alpha = 0$), so that the null space of P' and R will have dimension $\delta > d$. The additional solutions lead to complications in later steps. Thus suppose that $\hat{U}$ is an $M\times d$ matrix whose columns form an orthonormal basis of the column span of A, or an unbiased approximation thereof. It can e.g., be computed from an SVD of $R_e^{-1/2}\tilde{X}$. Instead of $\tilde{X}$, we now work with a rank reduced data matrix

$$\hat{X} := \hat{U}^*\tilde{X} .$$

We have $\hat{X} = \hat{U}^*(X+E)$ so that the noise on $\hat{X}$ has covariance

$$\hat{R}_e := \hat{U}^* R_e \hat{U} .$$

The algorithm then uses $\hat{R}$ based on $\hat{X}$, and $\hat{R}_n$ based on $\hat{R}_e$ and $\hat{X}$.

Note that it is not critical that $\hat{U}$ is an exact basis for A, as long as $T = \hat{U}^*A$ has full rank d: in that case the transformed problem $\hat{X} = TS + \hat{E}$ still allows to separate the sources. However, chosing $\mathrm{col}(\hat{U}) = \mathrm{col}(A)$ will optimally preserve the information on the signals while truncating $M-d$ dimensions of the noise.

White gaussian noise Assume that $\hat{U}$ contains the d dominant singular vectors of $\tilde{X}$, and let $\hat{\Sigma}$ be a diagonal matrix containing the corresponding singular values. For white gaussian noise, $\hat{R}_e$ is equal to

$$\hat{R}_e = \sigma^2 I .$$

We also have $\sum\hat{\mathbf{x}}\hat{\mathbf{x}}^* = \hat{\Sigma}^2$ so that

$$\hat{R}_n = \hat{\Sigma}^2\otimes I + I\otimes\hat{\Sigma}^2$$

Given data $\tilde{X}$ and noise covariance R_e, compute beamformer W

1. SVD: $R_e^{-1/2}\tilde{X} =: U\Sigma V^*$
   Rank reduction: $\hat{X} := \hat{U}^*\tilde{X}$, $\hat{R}_e := \hat{U}^* R_e \hat{U}$
   Construct P with rows $\text{vech}(\hat{\mathbf{x}}_k\hat{\mathbf{x}}_k^*)^T$
   QR fact.: $Q[\mathbf{1} \quad P] =: \begin{bmatrix} \sqrt{N} & \mathbf{p}^* \\ \mathbf{0} & P' \end{bmatrix}$
   $\hat{R}_n := J\left[\hat{\Sigma}^2 \otimes \hat{R}_e + \hat{R}_e^T \otimes \hat{\Sigma}^2\right] J^*$
   SVD: $\{\mathbf{y}_i'\} = \ker(P'\hat{R}_n^{-1/2})$
   $\mathbf{y}_i = \hat{R}_n^{-1/2}\mathbf{y}_i' \;\; (i = 1, \cdots, d)$
   $Y_i = \text{vech}^{-1}\mathbf{y}_i \;\; (i = 1, \cdots, d)$
2. Continue as in the usual ACMA [5]

Figure 1. Weighted ACMA

is diagonal. For $d = 2$, suppose $\hat{\Sigma} = \text{diag}[\sigma_1, \sigma_2]$, then

$$\hat{R}_n = \text{diag}[2\sigma_1^2,\; \sigma_1^2 + \sigma_2^2,\; \sigma_1^2 + \sigma_2^2,\; 2\sigma_2^2]\,.$$

This shows that the weighting is significant only if the singular values are unequal, i.e., for unequal source powers, or closely spaced sources.

Real processing

A hermitian symmetry is present:

$$\mathbf{y} = \bar{\mathbf{w}} \otimes \mathbf{w} = \text{vec}(\mathbf{w}\mathbf{w}^*)\,.$$

Instead of the 'vec$(\cdot)$' operator which stacks the columns, we can define a 'vech$(\cdot)$' operator, which essentially takes the real part of the above-diagonal entries, and the imaginary part of below-diagonal entries. This leads to the existence of a data independent unitary matrix J with a simple structure, such that

$$\text{vech}(\mathbf{w}\mathbf{w}^*) = J(\bar{\mathbf{w}} \otimes \mathbf{w}) \in \mathbb{R}\,.$$

The equation $P\mathbf{y} = \mathbf{1}$ is replaced by $(PJ^*)(J\mathbf{y}) = \mathbf{1}$, where PJ^* is real as well. Similarly, $\hat{R}$ is replaced by $J\hat{R}J^*$ and is real symmetric, and if we repeat the derivation of theorem 2, it follows that $\hat{R}_n$ is replaced by $J\hat{R}_n J^*$, and is also real symmetric. Note that if $\hat{R}_n$ is diagonal (as it is after prewhitening), then J has no effect on $\hat{R}_n$ and can be omitted.

The resulting algorithm is summarized in figure 1.

6. SIMULATIONS

Some performance results are shown in figure 2. In this simulation, we took a ULA$(\frac{\lambda}{2})$ consisting of $M = 4$ antennas, and $d = 3$ equal-power constant-modulus sources. In figure $2(a)$, we vary the number of samples N and the signal to noise ratio (SNR). The performance measure is the residual signal to interference ratio (SIR), which indicates how well the computed beamforming matrix W is an inverse of A.

Figure $2(b)$ shows the SIR for three sources with directions $[-\alpha, 0, \alpha]$, for varying α. The signal to noise ratio (SNR) was set at 10 dB, and we took $N = 200$ samples.

The plots show that the whitening removes the saturation of SIR as present in ACMA for large N, leading to substantial improvements for $N > 100$ and SNR between 0 and 25 dB. For SNR smaller than 0, the bias removal is ineffective because of our approximations. As seen in figure $2(b)$, the whitening is mostly useful if the singular values are sufficiently distinct, i.e., for small source separations.

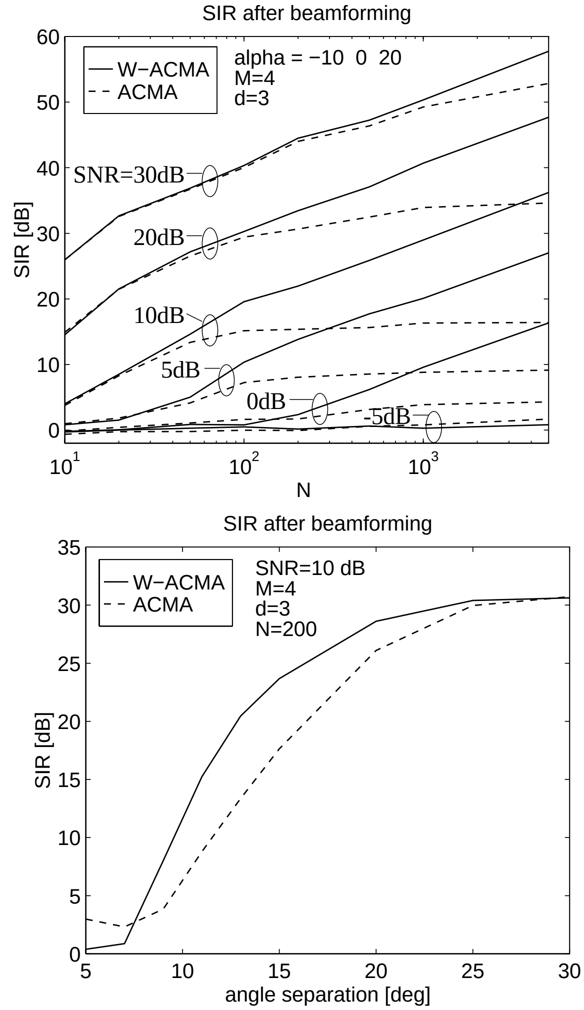

Figure 2. Performance of W-ACMA

References

[1] J.R. Treichler and B.G. Agee, "A new approach to multipath correction of constant modulus signals," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 31, pp. 459–471, Apr. 1983.

[2] J.R. Treichler and M.G. Larimore, "New processing techniques based on constant modulus adaptive algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 33, pp. 420–431, Apr. 1985.

[3] J.J. Shynk and R.P. Gooch, "The constant modulus array for cochannel signal copy and direction finding," *IEEE Trans. Signal Proc.*, vol. 44, pp. 652–660, Mar. 1996.

[4] C.B. Papadias and A.J. Paulraj, "A constant modulus algorithm for multiuser signal separation in presence of delay spread using antenna arrays," *IEEE Signal Proc. Letters*, vol. 4, pp. 178–181, June 1997.

[5] A.J. van der Veen and A. Paulraj, "An analytical constant modulus algorithm," *IEEE Trans. Signal Processing*, vol. 44, pp. 1136–1155, May 1996.

[6] A.L. Swindlehurst, M.J. Goris, and B. Ottersten, "Some experiments with array data collected in actual urban and suburban environments," in *IEEE workshop on Signal Proc. Adv. in Wireless Comm.*, (Paris), pp. 301–304, Apr. 1997.