Chapter 5

ALGEBRAIC CONSTANT MODULUS ALGORITHMS

Alle-Jan van der Veen

Delft University of Technology
Department of Electrical Engineering
2628 CD Delft, The Netherlands

allejan@cas.et.tudelft.nl

The constant modulus of many types of communication signals is a robust property for blind source separation and equalization. In this chapter, we discuss algebraic methods that use this constant modulus property to blindly compute separating beamformers from instantaneous mixtures of such sources, needing only a small number of observations.

5.1 Introduction

Constant modulus algorithms (CMAs) enjoy widespread popularity as methods for blind source separation and equalization of communication signals. As a typical application, consider a wireless scenario in which a number of users are broadcasting signals at the same frequency at the same time. The signals received at a base station will be some superposition of the transmitted sources. If the base station is equiped with multiple antennas, then it is likely that each antenna will receive a different combination of the signals. By linearly combining the antenna outputs, the objective is to separate the signals and to receive each of them while suppressing interference from the other signals. The task of the blind beamformer is to compute the proper linear combinations from the measured data only, without detailed knowledge of the signals or the channel.

Mathematically, the situation is described by the simple and well-known data model (after sampling and baseband conversion)

$$\mathbf{x}(k) = A\mathbf{s}(k) \tag{5.1.1}$$

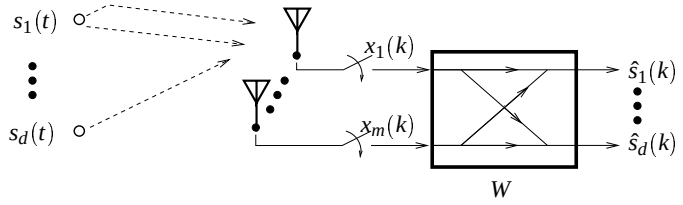

Figure 5.1. Blind beamforming scenario

where the vector $\mathbf{x}(k)$ is a stacking of the m antenna outputs $x_i(k)$ at discrete time k, $\mathbf{s}(k)$ is a stacking of the d source signals $s_i(k)$, and A is the array response matrix which describes the linear combinations of the signals as received by the antennas. This model is a reasonably accurate description for stationary propagation environments in which the multipath has only a short delay spread (as compared to the inverse of the signal bandwidths), so that no equalization is required. The beamforming problem is to find weight vectors $\mathbf{w}_i$, one for each source, such that $\mathbf{w}_i^*\mathbf{x}(k) = s_i(k)$ is equal to one of the original sources, without interference from the others. (* denotes a complex conjugate transpose.) Equivalently, we try to find A and then a pseudo-inverse of it such that $W^*A = I$. The columns of W are equal to the $\mathbf{w}_i$.

Although we will be concerned with blind beamforming, it is useful to note that a quite similar problem arises in the context of *blind equalization* of a single source observed through an unknown time-dispersive FIR channel. In that situation, the received signal $x(k)$ is a linear combination of shifts of the original source $s(k)$. By feeding $x(k)$ through a tapped delay line, we can construct a vector of received signals and we will arrive at the same model as (5.1.1), be it with more structure since $s_i(k) = s(k-i)$ and $x_i(k) = x(k-i)$. Another aspect that distinguishes blind equalization from blind beamforming is that in the latter we try to receive *all* independent sources.

Originally, most blind beamforming algorithms have been focusing on properties of A. For example, direction finding algorithms assume that the columns of A are vectors on the array manifold, each associated to a certain direction-of-arrival (DOA). By finding these directions, we obtain an estimate of A, and subsequently we can construct a beamformer W to separate the sources. This approach requires a calibrated array, and a scenario with very limited multipath propagation (since all DOAs have to be estimated).

A second class of approaches, more promising in the presence of unstructured multipath and useful in the context of blind equalization as well, exploits structural properties of the source vector that should hold and be reconstructed by the beamformer. One widely used property, and the property considered here, is the *constant modulus* of many communication signals (e.g. FM and PM in the analog domain, and FSK, PSK, 4-QAM for digital signals). For such signals, the amplitude $|s(k)|$ is a constant, typically normalized to 1, and all information is carried in the phase. A related but distinct property is the *finite alphabet* of digital signals, for example for a BPSK source $s(k) = \pm 1$. The idea of modulus restoral is to play with the weights of a beamformer $\mathbf{w}$ until the output $\hat{s}(k) = \mathbf{w}^*\mathbf{x}(k)$ has the same property,

$|\hat{s}(k)| = 1$, for all k. If that is the case, the output signal will be equal to one of the original sources [44].

Iterative CMAs A popular implementation of such property restoral algorithms is found by writing down a suitable cost function and minimizing it using stochastic gradient-descent techniques. For example, for a sample vector $\mathbf{x}$ we can consider as cost function the expected deviation of the squared modulus of the output signal $\mathbf{w}^*\mathbf{x}$ to a constant, say 1:

$$J_{\mathbf{w}} \;=\; \mathrm{E}(|\mathbf{w}^*\mathbf{x}|^2 - 1)^2 \,.$$

The corresponding so-called CMA(2,2) stochastic gradient-descent algorithm to find a minimizer $\mathbf{w}$ is given by the iteration

$$\mathbf{w}^{(k+1)} \;=\; \mathbf{w}^{(k)} - \mu \mathbf{x}\mathbf{x}^*\mathbf{w}^{(k)}\,(|\mathbf{w}^{(k)*}\mathbf{x}|^2 - 1)$$

where μ is a small step size. The vector $\mathbf{x}$ is usually taken to be equal to a sample vector $\mathbf{x}(k)$, in which case we obtain an adaptive LMS-type algorithm, but it is also possible to revisit the data. In any case, we need to select a suitable step size μ and an initial point $\mathbf{w}^{(0)}$ for the iteration. This type of ideas for modulus restoral has its roots in the work of Sato [38], Godard [17], and Treichler, Agee, and Larimore [44], [28], all for the purpose of blind equalization. See [26] for a recent review. The application of the CMA to blind beamforming is straightforward and was first considered in [45], [19] (see also [40]); a combined spatio-temporal CMA was proposed in [20].

A second type of algorithms are block-iterative. We consider a block of data $X = [\mathbf{x}(1), \cdots, \mathbf{x}(n)]$ and try to find $\mathbf{w}$ and $\mathbf{s}$ that minimize

$$\min_{\mathbf{w},\mathbf{s}} \|\mathbf{w}^*X - \mathbf{s}\|\,, \qquad \text{s.t. } |s(k)| = 1\,,\; k = 1, \cdots, n\,. \tag{5.1.2}$$

where $\|\cdot\|$ is the vector 2-norm. Optimizing jointly over $\mathbf{w}$ and $\mathbf{s}$ is hard. However, given $\mathbf{w}$, we can easily minimize over $\mathbf{s}$ with $\mathbf{w}$ fixed: the solution is to take $\mathbf{w}^*X$ and project it onto the set of constant modulus signals by dividing entrywise by the moduli, using the projection operator

$$\mathbf{P}_{\mathcal{CM}}(\mathbf{s}) := \left[\frac{s(1)}{|s(1)|}\,,\; \cdots,\; \frac{s(n)}{|s(n)|}\right] .$$

Similarly, if $\mathbf{s}$ is kept fixed, we can optimize over $\mathbf{w}$ and find the Least Squares solution $\mathbf{w}^* = \mathbf{s}X^\dagger$, where † denotes the Moore-Penrose pseudo-inverse. Given an initial point $\mathbf{w}^{(0)}$, we thus obtain the following two-step block-iteration:

$$\left[\begin{array}{lcl} \mathbf{s}^{(k+1)} & = & \mathbf{P}_{\mathcal{CM}}(\mathbf{w}^{(k)*}X) \\ \mathbf{w}^{(k+1)*} & = & \mathbf{s}^{(k+1)}X^\dagger \end{array}\right. \tag{5.1.3}$$

Note that $\mathbf{s}^{(k+1)} = \mathbf{P}_{\mathcal{CM}}(\mathbf{s}^{(k)}X^\dagger X)$ where $X^\dagger X$ is a projection onto the row span of X: the iteration is recognized as an Alternating Projection algorithm. It is known as the Gerchberg-Saxton algorithm (GSA), a well-established algorithm in the field of optics for solving the phase-retrieval problem [16]. It is closely connected to several variants of the CMA iteration (the OCMA [19] and the LSCMA [2]), except that it iterates on blocks rather than individual vectors.

Limitations of iterative CMAs A complication with the above iterative CMAs is that they find only a single weight vector, hence only a single signal will be recovered. It depends on the initialization of $\mathbf{w}^{(0)}$ to which one we converge to. This is sufficient in the context of equalization where all signals are equal up to a shift, but for blind beamforming, we would like to recover all independent signals. One idea is to redo the iteration with a range of initial points. The problem with this is that we might converge to the same solution: we have to verify that the resulting signals are indeed independent, and there is no good way to know that we have found all CM signals present in the data. Several solutions have been proposed for this. For example, once signal $s_1(k)$ is found, we can try to remove it from the data by estimating its corresponding response vector $\mathbf{a}_1$, and continue with the residual $\mathbf{x} - \hat{\mathbf{a}}_1\hat{\mathbf{s}}_1$. This is the strategy followed by the CM Array algorithm [40], [33], [27] and more recently in the context of CDMA by Parallel/Serial Interference Cancellation (PIC/SIC) algorithms, of which there are many variants. Another remedy is to augment the cost function with a term expressing independence [36], [35]. In both cases, several hundreds of samples are usually needed before convergence is satisfactory. (This number may be reduced if the same data is revisited.)

A further complication in finding all independent solutions is that, if the number of sensors is larger than the number of sources, there exist vectors $\mathbf{w}_0$ in the left null space of A such that $\mathbf{w}_0^* A = \mathbf{0}$. These vectors can be added to any solution $\mathbf{w}$ without changing the output signal. It is thus possible that independent beamforming vectors give rise to the same output signals, and hence it is not sufficient to require the independence of the $\mathbf{w}$.

Iterative CMAs are straightforward to implement and computationally of modest complexity. They can however converge slowly, with unpredictable convergence speed, and the recovering of all independent sources remains a problem. It is thus interesting to note that the problem admits an elegant and algebraic solution. This is the Algebraic CMA (ACMA), and the topic of this chapter.

ACMA The ACMA was introduced in [50]. In this paper, it was noted that in the noise-free case the collection of d beamformers for all individual signals can be computed exactly and algebraically, as the solution of a generalized eigenvalue problem. Only a limited number of samples is needed: for d signals, it is sufficient to have $n > d^2$ samples. Also the number of CM signals can be detected.

The algorithm is derived by setting up the equations for the weight vector $\mathbf{w}$ such that $\mathbf{w}^* X$ is a CM signal. This gives n quadratic equations in the entries of $\mathbf{w}$. Using properties of Kronecker products, the problem can be formulated as an overdetermined linear system subject to a quadratic structural constraint. The linear system can be solved, and leads to a d-dimensional basis of solutions on which we have to apply the structural constraint. This can then be formulated as a generalization of an eigenvalue problem: the simultaneous diagonalization of a number of matrices.

An illustration of the performance of ACMA on experimental data [50] is shown in figure 5.2. The sources are six FM modulated analog speech/music signals, occupying the same subband of 25 kHz in the 900 Mhz band, and broadcast at different locations at a rooftop. The receiving antenna array consists of $m = 6$ omnidirectional antennas, with a maximal baseline of 2.5m. The signal-to-noise ratio is around 17 dB per antenna per

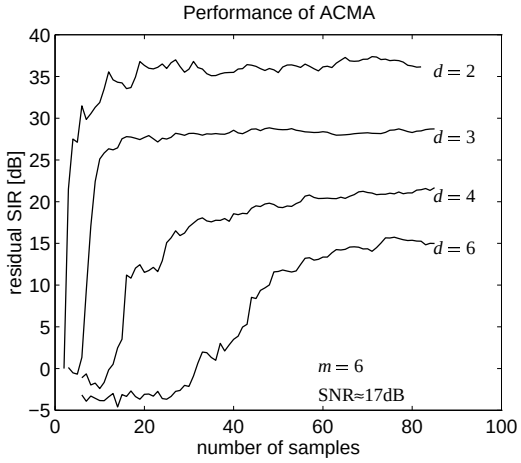

Figure 5.2. Residual signal-to-interference ratio after blind beamforming of a mixture of d signals using ACMA.

source, and the sources have roughly equal powers. The figure shows the worst signal-to-interference ratio (SIR) among the signals after beamforming, as a function of the number of samples that have been used, and for a varying number of sources. It is seen that only a small number of samples are required (order $2d^2$ or so) to give a good separation of all d sources, even though some of the sources are only spaced by 1.5°.

Outline of the chapter Section 5.3 provides a compact derivation of the original ACMA. We then analyze the noise-free properties (section 5.4) and asymptotic properties of the algorithm, discuss its connection to the related JADE algorithm, and show that ACMA converges to the Wiener solution (section 5.6). We also derive a Weighted ACMA which approximately converges to the zero-forcing solution (section 5.7), and compare ACMA, WACMA and JADE in simulations (section 5.9). Section 5.8 considers a specialization to binary sources. Finally, section 5.10 goes into more details on simultaneous diagonalization algorithms.

5.2 Preliminaries

Data model and assumptions

Starting from the data model $\mathbf{x}_k = A\mathbf{s}_k$, let us assume that we have collected n sample vectors. If we store the samples in an $m \times n$ matrix $X = [\mathbf{x}_1, \cdots, \mathbf{x}_n]$, then we obtain that X has a factorization

$$X = AS \tag{5.2.1}$$

where the rows of $S \in \mathbb{C}^{d \times n}$ contain the samples of the source signals. All sources are assumed to be constant modulus, $|S_{ij}| = 1$. Note that the source powers are absorbed in A.

In the blind signal separation scenario, both A and S are unknown, and the objective is, given X, to find the factorization $X = AS$ such that $|S_{ij}| = 1$. Alternatively, we try to find a beamforming matrix $W \in \mathbb{C}^{m \times d}$ of full row rank d such that $S = W^* X$. For this left inverse W^* of A to exist, it immediately follows that we need A to have full rank d, and $m \geq d$. We also require the rows of S to be linearly independent. As will become clear in the derivation of the algorithm, we will not really need stochastic independence of the sources, but some form of persistence of excitation should hold. The algorithm will also require that $n \geq d^2$.

As in other blind source separation problems, it is clear that we can recover A and S only up to a permutation of the sources, and up to a complex unimodular diagonal (since the initial phases of the sources does not follow from the data, these factors can be exchanged between A and S). This is precisely the indeterminacy of the eigenvalue problem from which the beamformers will be derived.

In the presence of additive noise, we write $\tilde{\mathbf{x}}_k = A\mathbf{s}_k + \mathbf{n}_k$, or

$$\tilde{X} = AS + N. \tag{5.2.2}$$

We use the tilde ($\tilde{}$) to denote variables derived from the noisy data. The noise is assumed to be additive, temporally i.i.d., zero mean, circularly symmetric, with finite covariance $\mathrm{E}(\mathbf{n}\mathbf{n}^*)$ and fourth-order moments, and independent from the sources.

Identifiability We will assume that the problem is essentially *identifiable*, i.e., that for a given matrix X of size $m \times n$, we can find a factorization $X = AS$ ($|S_{ij}| = 1$) which is unique up to the above-mentioned indeterminacies. Minimal conditions that guarantee this identifiability are not completely known. For $n \to \infty$, it is sufficient to have A full column rank and S generated by statistically independent signals: in this case the CM cost function has unique global minima corresponding to separating beamformers [44]. By counting the number of equations and unknowns (a not completely convincing argument), it is motivated in [50] that identifiability is expected for $n \geq 2d$ and sufficiently exciting signals. The ACMA requires $m \geq d$ and $n \geq d^2$. Finally, for BPSK signals (and other discrete alphabets), it was established in [42] that the factorization is essentially unique once all constellation vectors have been received.

Additional notation Overbar ($\bar{}$)denotes complex conjugation, ^{T} is the matrix transpose, * the matrix complex conjugate transpose, $\dagger$ the matrix pseudo-inverse (Moore-Penrose inverse). $\mathbf{0}$ 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:

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

$\mathrm{vec}(A)$ indicates a stacking of the columns of a matrix A into a vector. Notable properties

are:

$$\begin{array}{lcl} \mathrm{vec}(ABC) & = & (C^T \otimes A)\mathrm{vec}(B) \\ (A\otimes B)(C\otimes D) & = & AC\otimes BD \\ \mathrm{vec}(\mathbf{ab}^*) & = & \bar{\mathbf{b}}\otimes\mathbf{a}\,. \end{array}$$

For a $d\times d$ *Hermitian* matrix Y, an alternative vectorization that leads to a real-valued result is provided by the $\mathrm{vech}(\cdot)$ operator, defined as

$$\mathbf{y}' = \mathrm{vech}(Y) \quad \Leftrightarrow \quad \mathbf{y}'_{(i-1)d+j} = \begin{cases} y_{ii}\,, & i=j \\ \mathrm{real}(y_{ij})\sqrt{2}, & i<j \\ \mathrm{imag}(y_{ji})\sqrt{2}, & i>j \end{cases} \qquad i,j=1,\cdots,d\,. \tag{5.2.3}$$

This vectorization follows from $\mathrm{vec}(Y)$ by combining entries y_{ij} and $y_{ji}=\overline{y}_{ij}$ to produce the real and imaginary part of y_{ij}. The definition is such that $\|\mathrm{vech}(Y)\| = \|\mathrm{vec}(Y)\|$ for any Hermitian Y, and hence there exists a unitary matrix J such that $\mathrm{vech}(Y) = J\mathrm{vec}(Y)$.

Zero-Forcing and Wiener receivers

Before we look at blind beamforming algorithms, it is good to recall what beamforming solutions we would prefer if either A or S is known. In the noisefree case, we have a data model $X=AS$, and we would like a beamformer W such that $W^*X=S$. If A is known, then we set

$$W^* = A^\dagger\,, \qquad S = W^*X$$

whereas if S is known, for example because of a training segment, then we take

$$W^* = SX^\dagger\,, \qquad A = (W^*)^\dagger\,.$$

In both cases, we obtain a beamformer which exactly cancels all interference, i.e., $W^*A=I$.

In the presence of additive noise, we have $\tilde{X}=AS+N$. Two types of linear least-squares (LS) miminization problems can now be considered: either based on minimizing the modeling error,

$$\min_{A,S} \|\tilde{X}-AS\|_F^2 \quad \text{s.t. conditions on } (A,S)\,, \tag{5.2.4}$$

or based on minimizing the output error,

$$\min_{W,S} \|W^*\tilde{X}-S\|_F^2 \quad \text{s.t. conditions on } (W,S)\,. \tag{5.2.5}$$

The conditions on A or W and S are those posed by the blind identification problem at hand. In the CM case, we have the condition that all $|S_{ij}|=1$.

The minimization problems are straightforward to solve if either A or S is known. In the first formulation, if S is known,

$$\hat{A} = \underset{A}{\mathrm{argmin}} \parallel \tilde{X}-AS\parallel_F^2 = \tilde{X}S^\dagger \tag{5.2.6}$$

Asymptotically for zero mean noise independent of the sources, this gives $\hat{A} \to A$: we converge to the true A-matrix. For a known A, the estimate of S follows from

$$\hat{S} = \underset{S}{\operatorname{argmin}} \parallel \tilde{X} - AS \parallel_F^2 = A^\dagger X \tag{5.2.7}$$

with corresponding beamformer $W = (A^\dagger)^*$. This is also known as the zero-forcing (ZF) beamformer, because $W^*A = I$: all interfering sources are cancelled. The ZF beamformer maximizes the Signal-to-Interference power Ratio (SIR) at the output.

The second optimization problem minimizes the difference of the output signals to S:

$$W^* = \underset{W}{\operatorname{argmin}} \parallel W^*\tilde{X} - S \parallel_F^2 = S\tilde{X}^\dagger . \tag{5.2.8}$$

Note that $\tilde{X}^\dagger = \tilde{X}^*(\tilde{X}\tilde{X}^*)^{-1}$, so that

$$W^* = \tfrac{1}{n} S\tilde{X}^* (\tfrac{1}{n}\tilde{X}\tilde{X}^*)^{-1} \quad = \quad \tilde{R}_{xs}^* \tilde{R}_x^{-1} .$$

$\tilde{R}_x := \frac{1}{n}\tilde{X}\tilde{X}^*$ is the sample data covariance matrix, and $\tilde{R}_{xs} := \frac{1}{n}(\tilde{X}S^*)$ converges to A. Hence asymptotically

$$W \to \tilde{R}_x^{-1} A$$

which is recognized as the Linear Minimum Mean Square Error (LMMSE) or Wiener receiver. This beamformer is known to maximize the Signal-to-Interference-plus-Noise Ratio (SINR) at the output. Since it does not cancel all interference, $W^*A \neq I$, the output source estimates are not unbiased. However, it produces estimates of S with minimal deviation, which is often more relevant.

Block-iterative CMAs From the preceding equations, it is straightforward to derive other block-iterative CMAs. Indeed, given either an initial estimate $A^{(0)}$ or $W^{(0)}$, we can follow the fixed-point iterations corresponding to (5.2.6), (5.2.7) and (5.2.8)

$$\left[\begin{array}{lcl} S^{(k)\prime} & = & A^{(k)\dagger}\tilde{X} \\ S^{(k+1)} & = & \mathbf{P}_{\mathcal{CM}}(S^{(k)\prime}) \\ A^{(k+1)} & = & \tilde{X}S^{(k+1)\dagger} \end{array}\right] \quad \text{or} \quad \left[\begin{array}{lcl} S^{(k)\prime} & = & W^{(k)*}\tilde{X} \\ S^{(k+1)} & = & \mathbf{P}_{\mathcal{CM}}(S^{(k)\prime}) \\ W^{(k+1)*} & = & S^{(k+1)}\tilde{X}^\dagger \end{array}\right] \tag{5.2.9}$$

Given sufficiently accurate initial points, the former algorithm will converge to the ZF solution. The latter algorithm should converge to the Wiener solution, and is an extension of the alternating projection algorithm in (5.1.3). Computationally, it is more attractive than the former iteration, because $\tilde{X}$ has to be inverted only once. However, the algorithm has the problem that it does not guarantee that the rows of S will be independent: it is possible that we will find the same signal several times. This is a general problem with the formulation $\min \|W^*\tilde{X} - S\|_F$.

Whitening and rank reduction

In the noise-free case with less sources than sensors, $X = AS$ is rank deficient: its rank is d (the number of signals) rather than m (the number of sensors). As a consequence, once

we have found a beamformer $\mathbf{w}$ such that $\mathbf{w}^* X = \mathbf{s}$, one of the source signals, then we can add any vector $\mathbf{w}_0$ such that $\mathbf{w}_0^* X = \mathbf{0}$ to $\mathbf{w}$, and obtain the same output. The beamforming solutions are not unique.

The desired beamforming solutions are all in the column span of A. Indeed, any component orthogonal to this span will not contribute at the output. The most easy way to ensure that our solutions will be in this span is by performing a dimension-reducing prefiltering. Let F be any $m \times d$ matrix such that $\text{span}(F) = \text{span}(A)$. Then all beamforming matrices W in the column span of A are given by

$$W = FT$$

where T is a $d \times d$ matrix, nonsingular if the beamformers are linearly independent. The prefiltered data matrix is $\underline{X} := F^* X$. We will use the underscore ($\underline{\ }$) to denote prefiltered variables. Thus, the prefiltered noisy data matrix is

$$\underline{\tilde{X}} := F^* \tilde{X}$$

with structure

$$\underline{\tilde{X}} = \underline{A} S + \underline{N}, \qquad \text{where} \quad \underline{A} := F^* A, \ \underline{N} := F^* N.$$

$\underline{\tilde{X}}$ has only d channels, and is such that $W^* \tilde{X} = T^* \underline{X}$. Thus, the columns of T are d-dimensional beamformers on the prefiltered data $\underline{\tilde{X}}$, and for any choice of T the columns of the effective beamformer W are all in the column span of A, as desired.

To describe the column span of A, introduce the "economy-size" singular value decomposition of A,

$$A = U_A \Sigma_A V_A$$

where we take $U_A : m \times d$ with orthonormal columns, $\Sigma_A : d \times d$ diagonal containing the nonzero singular values of A, and $V_A : d \times d$ unitary. Also let $U_A^\perp$ be the orthonormal complement of U_A. The columns of U_A are an orthonormal basis of the column span of A. The point is that even if A is unknown, U_A can be estimated from the data, as described below.

We assume that the noise is spatially white, with covariance matrix $\sigma^2 I$. Let $\tilde{R}_x = \frac{1}{n} \tilde{X} \tilde{X}^*$ be the noisy sample data covariance matrix, with eigenvalue decomposition

$$\tilde{R}_x \ = \ U \Sigma^2 U^* . \tag{5.2.10}$$

Here, U is $m \times m$ unitary, and Σ is $m \times m$ diagonal. Let us collect the d largest eigenvalues into a diagonal matrix $\hat{\Sigma}^2$, and collect the corresponding d eigenvectors into $\hat{U}$. Asymptotically,[1] $\tilde{R}_x$ satisfies $\tilde{R}_x \doteq A A^* + \sigma^2 I$, with eigenvalue decomposition

$$\tilde{R}_x \ \doteq \ U_A \Sigma_A^2 U_A^* + \sigma^2 I \ = \ U_A (\Sigma_A^2 + \sigma^2 I) U_A^* + \sigma^2 U_A^\perp U_A^{\perp *} . \tag{5.2.11}$$

It follows that $\hat{U} \hat{\Sigma}^2 \hat{U}^* \doteq U_A (\Sigma_A^2 + \sigma^2 I) U_A^*$, so that $\hat{U}$ is an asymptotically unbiased estimate of U_A.

Even if we choose F to have the column span of $\hat{U}$, there is freedom left. It will follow later in section 5.5 that a natural choice will be to combine the dimension reduction with a

[1] We use $\doteq$ to denote asymptotic equality.

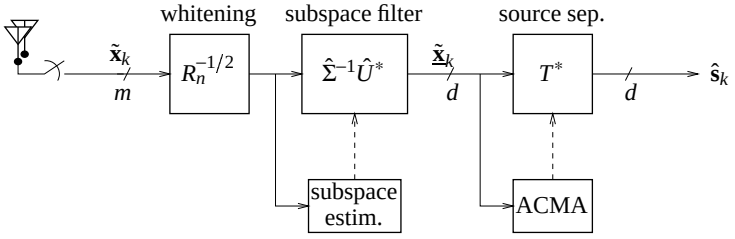

Figure 5.3. Blind beamforming prefiltering structure

whitening of the data covariance matrix, i.e., such that $\underline{\tilde{R}}_x := \frac{1}{n}\underline{\tilde{X}}\underline{\tilde{X}}^*$ becomes unity: $\underline{\tilde{R}}_x = I$. This is achieved if we define F as

$$F = \hat{U}\hat{\Sigma}^{-1}. \tag{5.2.12}$$

If the noise is colored with covariance matrix $\sigma^2 R_n$, where we know R_n but perhaps not the noise power σ^2, then we first whiten the noise by computing $R_n^{-1/2}\tilde{X}$, and continue as in the white noise case. The resulting prewhitening/dimension reducing filter is

$$F = R_n^{-1/2}\hat{U}\hat{\Sigma}^{-1}.$$

The structure of the resulting beamformer is shown in figure 5.3. Incidentally, note that the fact that with finite samples the column span of F only approximates that of A does not introduce a bias: as long as $\underline{A} = F^*A$ has full rank d it can be inverted by some T^*, and a perfect zero-forcing beamformer still exists.

Sometimes slightly different prefilters are used. In some blind source separating algorithms which try to estimate A, the SVD of A is introduced as $A = U_A\Sigma_A V_A$, and it is remarked that the factors U_A and Σ_A can be estimated from the eigendecomposion of the data covariance matrix, since asymptotically (5.2.11) holds: $U_A \doteq \hat{U}$ and $\Sigma_A \doteq (\hat{\Sigma}^2 - \sigma^2 I)^{1/2}$. For this it is also necessary to have an estimate of the noise power, but if $d < m$ it can be estimated from the smallest eigenvalues of $\tilde{R}_x$. If we thus take the prefiltered data matrix to be

$$\underline{\tilde{X}} \;=\; F^*\tilde{X}\,, \qquad F = \hat{U}(\hat{\Sigma}^2 - \sigma^2 I)^{-1/2} \tag{5.2.13}$$

then asymptotically $\underline{\tilde{X}} \doteq V_A S + \underline{N}$, and the search for A can be restricted to the search for a unitary matrix V_A. For finite data, this asymptotic result is not yet valid, and the restriction leads to a bias.

5.3 Derivation of the ACMA

Outline

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

$$\mathbf{w}^*X = \mathbf{s}, \qquad \text{such that} \quad |s_k|^2 = 1 \quad (k = 1, \cdots, n).$$

Let $\mathbf{x}_k$ be the k-th column of X. By substitution, we find

$$\mathbf{w}^*(\mathbf{x}_k\mathbf{x}_k^*)\mathbf{w} = 1\,, \qquad k = 1,\cdots,n\,. \tag{5.3.1}$$

Thus, there are n equations, and each equation is quadratic in the entries of $\mathbf{w}$. This is not an easy problem, but it becomes more manageable if we write the unknowns on one side. This is possible using properties of Kronecker products, in particular

$$\mathbf{w}^*(\mathbf{x}_k\mathbf{x}_k^*)\mathbf{w} = (\bar{\mathbf{x}}_k \otimes \mathbf{x}_k)^*(\bar{\mathbf{w}} \otimes \mathbf{w})$$

Thus define

$$P := [\bar{X} \circ X]^* = \begin{bmatrix} (\bar{\mathbf{x}}_1 \otimes \mathbf{x}_1)^* \\ \vdots \\ (\bar{\mathbf{x}}_n \otimes \mathbf{x}_n)^* \end{bmatrix}$$

($P : n \times d^2$). Then (5.3.1) is equivalent to finding all $\mathbf{w}$ that satisfy

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

We have converted the problem into a linear system of equations, subject to a quadratic constraint. The linear system is overdetermined once $n \geq d^2$, and we will assume that this is the case.

In general outline, the ACMA technique solves this problem by the following steps:

1. *First solve the linear system $P\mathbf{y} = \mathbf{1}$.* Note that there are several independent solutions to the linear system. Indeed, if we have d sources, then there exist at least d solutions $\bar{\mathbf{w}}_i \otimes \mathbf{w}_i$ ($i = 1,\cdots,d$), one for each source. But also a linear combination of these solutions
$$\mathbf{y} = \lambda_1(\bar{\mathbf{w}}_1 \otimes \mathbf{w}_1) + \cdots + \lambda_d(\bar{\mathbf{w}}_d \otimes \mathbf{w}_d)$$
(scaled such that $\sum\lambda_i = 1$) will solve $P\mathbf{y} = \mathbf{1}$. Thus, if we select an arbitrary basis $\{\mathbf{y}_1,\cdots,\mathbf{y}_d\}$ of independent solutions of the linear system $P\mathbf{y} = \mathbf{1}$, we cannot expect to have found the desired structured solutions $\bar{\mathbf{w}}_i \otimes \mathbf{w}_i$, but rather unknown linear combinations of these.

2. *Decouple:* find a structured basis $\{\bar{\mathbf{w}}_1 \otimes \mathbf{w}_1,\cdots,\bar{\mathbf{w}}_d \otimes \mathbf{w}_d\}$ that spans the same linear subspace as $\{\mathbf{y}_1,\cdots,\mathbf{y}_d\}$. Since
$$\bar{\mathbf{w}}_i \otimes \mathbf{w}_i = \mathrm{vec}(\mathbf{w}_i\mathbf{w}_i^*)$$
we can associate to each structured basis vector a rank-1 hermitian matrix. Our problem is thus to split a given subspace into its "rank-1 components".

   To solve this problem, note that in the same way we can associate to each $\mathbf{y}_i$ a $d \times d$ matrix Y_i such that $\mathrm{vec}(Y_i) = \mathbf{y}_i$. Since each $\mathbf{y}_i$ is in the span of the rank-1 components, each Y_i is an unknown linear combination of the rank-1 matrices:

$$\left\{\begin{array}{rcl} Y_1 &=& \lambda_{11}\mathbf{w}_1\mathbf{w}_1^* + \cdots + \lambda_{1d}\mathbf{w}_d\mathbf{w}_d^* \\ &\vdots& \\ Y_d &=& \lambda_{d1}\mathbf{w}_1\mathbf{w}_1^* + \cdots + \lambda_{dd}\mathbf{w}_d\mathbf{w}_d^* \end{array}\right. \quad\Leftrightarrow\quad \left\{\begin{array}{rcl} Y_1 &=& W\Lambda_1 W^* \\ &\vdots& \\ Y_d &=& W\Lambda_d W^* \end{array}\right. \tag{5.3.2}$$

where $\Lambda_i = \mathrm{diag}[\lambda_{i1}, \cdots, \lambda_{id}]$ and $W = [\mathbf{w}_1, \cdots, \mathbf{w}_d]$.

This problem is known as a *joint diagonalization problem* (by congruence), since all Y_i can be diagonalized into the Λ_i by the same matrix W. It is a generalization of the standard eigenvalue decomposition problem, and can be solved. For example, if Y_2 is invertible, then

$$Y_1 Y_2^{-1} = W(\Lambda_1 \Lambda_2^{-1}) W^{-1}$$

and if the eigenvalues are distinct, we can find W and hence all beamformers in one shot as the eigenvectors of $Y_1 Y_2^{-1}$. Numerically, it is better to consider all Y_i and to avoid inversions. Details and algorithms are in section 5.10.

3. In considering the $\mathbf{y}_i$ as a basis of a linear subspace (unconstrained), we have lost the correct scaling of the $\mathbf{w}_i$. Rather than constraining the Λ_i, this is more easily fixed by scaling each solution such that the average output power

$$\frac{1}{n}\sum_{k=1}^{n} |(s_i)_k|^2 \;=\; \frac{1}{n}\sum_{k=1}^{n} \mathbf{w}_i^* \mathbf{x}_k \mathbf{x}_k^* \mathbf{w}_i \;=\; \mathbf{w}_i^* \left(\frac{1}{n}\sum_{k=1}^{n} \mathbf{x}_k \mathbf{x}_k^* \right) \mathbf{w}_i \tag{5.3.3}$$

is equal to 1.

A crucial aspect of the above technique is that the basis $\{\mathbf{y}_i\}$ should not contain other components than the desired $\{\bar{\mathbf{w}}_i \otimes \mathbf{w}_i\}$, otherwise we cannot pose the problem as a joint diagonalization. For this, it is essential that there are precisely d linearly independent solutions to $P\mathbf{y} = \mathbf{1}$ and no additional spurious solutions. The situation is analyzed in section 5.4. As is shown in that section, two cases where the number of solutions will be too large are

– If X is rank deficient, e.g., because the number of sensors is larger than the number of sources. This is simply treated by a prewhitening combined with a dimension reduction, as we discussed in section 5.2.

– Additional rank-2 components occur for pairs of BPSK-type signals. This situation has to be ruled out. However, if all signals are known to be of this type, the algorithm can be modified (section 5.8).

Next, we go into some more details on various aspects of this algorithm.

Solving the linear system

We will now show how a basis of solutions for $P\mathbf{y} = \mathbf{1}$ can be constructed. The purpose of this is also to introduce some notation to be used in later sections. Let Q be any unitary matrix such that $Q\mathbf{1} = \sqrt{n}\left[\begin{smallmatrix}1\\ \mathbf{0}\end{smallmatrix}\right]$. Then apply Q to $[\mathbf{1}\ \ P]$:

$$Q[\mathbf{1} \quad P] =: \sqrt{n} \begin{bmatrix} 1 & \mathbf{p}^* \\ \mathbf{0} & G \end{bmatrix}, \tag{5.3.4}$$

(In practice, we would compute a QR factorization of $[\mathbf{1}\ \ P]$.) Then

$$P\mathbf{y} = \mathbf{1} \quad \Leftrightarrow \quad Q[\mathbf{1} \quad P] \begin{bmatrix} -1 \\ \mathbf{y} \end{bmatrix} = \mathbf{0} \quad \Leftrightarrow \quad \begin{cases} \mathbf{p}^*\mathbf{y} &= 1 \\ G\mathbf{y} &= \mathbf{0} \end{cases}$$

Given data $\tilde{X}$ and noise covariance R_n, compute beamformers W and output $S = W^*\tilde{X}$:

1. SVD: $R_n^{-1/2}\tilde{X} =: U\Sigma V^*$
   Prewhitening/dimension reduction: $\underline{\tilde{X}} := \hat{\Sigma}^{-1}\hat{U}^* R_n^{-1/2}\tilde{X}$
   Construct P with rows $\mathrm{vech}(\underline{\tilde{\mathbf{x}}}_k\underline{\tilde{\mathbf{x}}}_k^*)^T$
   QR fact.: $Q[\mathbf{1} \quad P] =: \sqrt{n}\begin{bmatrix} 1 & \mathbf{p}^* \\ \mathbf{0} & G \end{bmatrix}$
   SVD of G, set $\{\mathbf{y}_i'\}$ to the smallest d right singular vectors.
2. $Y_i = \mathrm{vech}^{-1}\mathbf{y}_i' \ (i = 1, \cdots, d)$
   Find T to jointly diagonalize Y_i as $Y_i = T\Lambda_i T^* \ (i = 1, \cdots, d)$
3. Scale each column of T to norm 1.
   Set $W = R_n^{-1/2}\hat{U}\hat{\Sigma}^{-1}T$ and $S = T^*\underline{\tilde{X}}$

Figure 5.4. Summary of ACMA

Thus, the problem of finding all solutions to $P\mathbf{y} = \mathbf{1}$ is effectively replaced by the second equation which says that $\{\mathbf{y}_i\}$ is a basis for the null space of the matrix G. It can be conveniently found from an SVD of G. The first equation fixes a scaling for each $\mathbf{y}_i$.

By squaring (5.3.4), we obtain explicit expressions for $\mathbf{p}$ and $C := G^*G$ that will be useful later:

$$\begin{aligned} \mathbf{p} &= \frac{1}{n}P^*\mathbf{1} = \frac{1}{n}\sum \bar{\mathbf{x}}_k \otimes \mathbf{x}_k \\ C := G^*G &= \frac{1}{n}P^*P - \mathbf{p}\mathbf{p}^* \\ &= \frac{1}{n}\sum (\bar{\mathbf{x}}_k \otimes \mathbf{x}_k)(\bar{\mathbf{x}}_k \otimes \mathbf{x}_k)^* - \left[\frac{1}{n}\sum \bar{\mathbf{x}}_k \otimes \mathbf{x}_k\right]\left[\frac{1}{n}\sum \bar{\mathbf{x}}_k \otimes \mathbf{x}_k\right]^* . \end{aligned} \tag{5.3.5}$$

The former expression shows that (for $\mathbf{y} = \bar{\mathbf{w}} \otimes \mathbf{w}$)

$$\mathbf{p}^*\mathbf{y} = \left(\frac{1}{n}\sum \bar{\mathbf{x}}_k \otimes \mathbf{x}_k\right)^* \mathbf{y} = \mathbf{w}^*\left(\frac{1}{n}\sum \mathbf{x}_k\mathbf{x}_k^*\right)\mathbf{w}.$$

Thus, the condition $\mathbf{p}^*\mathbf{y} = 1$ is equal to the condition in (5.3.3) in the last step of the algorithm outline, where the average output power of the beamformer is fixed to 1. The second step (joint diagonalization) then needs to be concerned only with decoupling an arbitrary scaled basis of the null space of G, or equivalently that of $C = G^*G$.

Real processing A computational aspect is the following. Note that a hermitian symmetry is present:

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

Thus, $\bar{\mathbf{w}} \otimes \mathbf{w}$ contains a redundancy which can be removed, leading to computational savings. Instead of the 'vec$(\cdot)$' operator which stacks the columns of an arbitrary matrix, we can define a real-valued 'vech$(\cdot)$' operator acting on hermitian matrices, which essentially takes the real part of the above-diagonal entries, and the imaginary part of below-diagonal entries (see the definition in equation (5.2.3)). The definition is such that there exists a data-independent and unitary matrix J such that $\text{vech}(Y) = J\text{vec}(Y)$ for any Hermitian matrix Y. Thus,

$$\text{vech}(\mathbf{w}\mathbf{w}^*) = J(\bar{\mathbf{w}} \otimes \mathbf{w}) \in \mathbb{R}^{d^2}$$

is real-valued. The equation $P\mathbf{y} = \mathbf{1}$ is now replaced by $(PJ^*)(J\mathbf{y}) = \mathbf{1}$, where PJ^* is real as well: its rows are given by $\text{vech}(\tilde{\mathbf{x}}\tilde{\mathbf{x}}^*)^T$. Eventually, we find a real-valued basis $\{\mathbf{y}_i'\}$, at which point we can set $\mathbf{y}_i = J^*\mathbf{y}_i'$. The corresponding matrices Y_i are then hermitian by construction.

Algorithm The resulting algorithm is summarized in figure 5.4. In comparison to the outline, additional ingredients are the prefiltering, for which an SVD of the data matrix $\tilde{X}$ is needed, and the real processing (using $\text{vech}(\tilde{\mathbf{x}}_k\tilde{\mathbf{x}}_k^*)$ rather than $\bar{\tilde{\mathbf{x}}}_k \otimes \tilde{\mathbf{x}}_k$). The motivation for using a prefilter which *whitens* the data covariance matrix follows from an analysis of the algorithm in the presence of noise, and will be given in section 5.5. The joint diagonalization step is described in detail in section 5.10.

5.4 Analysis of the noisefree case

The analysis of ACMA in the noisefree case can be limited to an analysis of the solutions of $G\mathbf{y} = \mathbf{0}$. If all solutions are in the subspace spanned by $\{\bar{\mathbf{w}}_i \otimes \mathbf{w}_i;\ i = 1, \cdots, d\}$, then the joint diagonalization step is able to separate an arbitrary basis of the null space into its rank-1 components, and we recover the true beamformers. Rather than looking at the null space of G, it will be more convenient to look at the null space of $C := G^*G$. This is of course equivalent, but the analysis of the structure of C extends more easily to an asymptotic analysis in the noisy case, later in section 5.6.

As was derived in (5.3.5),

$$C = G^*G = \frac{1}{n}\sum(\bar{\mathbf{x}}_k \otimes \mathbf{x}_k)(\bar{\mathbf{x}}_k \otimes \mathbf{x}_k)^* - \frac{1}{n}\left[\sum \bar{\mathbf{x}}_k \otimes \mathbf{x}_k\right]\frac{1}{n}\left[\sum \bar{\mathbf{x}}_k \otimes \mathbf{x}_k\right]^*.$$

With $\mathbf{x}_k = A\mathbf{s}_k$, we obtain

$$C = [\bar{A} \otimes A]C_s[\bar{A} \otimes A]^*$$

where

$$C_s := \frac{1}{n}\sum(\bar{\mathbf{s}}_k \otimes \mathbf{s}_k)(\bar{\mathbf{s}}_k \otimes \mathbf{s}_k)^* - \frac{1}{n}\left[\sum \bar{\mathbf{s}}_k \otimes \mathbf{s}_k\right]\frac{1}{n}\left[\sum \bar{\mathbf{s}}_k \otimes \mathbf{s}_k\right]^*.$$

C is positive semidefinite, because it is constructed as $C = G^*G$. Hence, the null space of C has two components: the null space of $[\bar{A} \otimes A]^*$, plus vectors $\mathbf{y}$ such that $[\bar{A} \otimes A]^*\mathbf{y}$ is a vector in the null space of C_s. The purpose of prefiltering with dimension reduction is to remove the former solutions beforehand, by replacing A by a square full rank matrix $\underline{A}$. In that case also $\bar{\underline{A}} \otimes \underline{A}$ is square full rank, with an empty null space. Thus, the interesting part

is the analysis of the null space of C_s, which is only dependent on the signals, not on their directions.

For the sake of exposition, we specialize C_s for the case of two CM signals, $s_1(k)$ and $s_2(k)$. Define

$$\begin{aligned} \rho &:= \frac{1}{n}\sum s_1(k)\bar{s_2}(k) \\ q &:= \frac{1}{n}\sum [s_1(k)]^2[\bar{s_2}(k)]^2 . \end{aligned}$$

Then (suppressing the time index)

$$\begin{aligned} C_s &= \frac{1}{n}\sum_k \begin{bmatrix} \bar{s_1}s_1 \\ \bar{s_1}s_2 \\ \bar{s_2}s_1 \\ \bar{s_2}s_2 \end{bmatrix} \begin{bmatrix} s_1\bar{s_1} & s_1\bar{s_2} & s_2\bar{s_1} & s_2\bar{s_2} \end{bmatrix} \\ &\quad - \frac{1}{n}\sum_k \begin{bmatrix} \bar{s_1}s_1 \\ \bar{s_1}s_2 \\ \bar{s_2}s_1 \\ \bar{s_2}s_2 \end{bmatrix} \cdot \frac{1}{n}\sum_k \begin{bmatrix} s_1\bar{s_1} & s_1\bar{s_2} & s_2\bar{s_1} & s_2\bar{s_2} \end{bmatrix} \\ &= \begin{bmatrix} 1 & \rho & \bar{\rho} & 1 \\ \bar{\rho} & 1 & \bar{q} & \bar{\rho} \\ \rho & q & 1 & \rho \\ 1 & \rho & \bar{\rho} & 1 \end{bmatrix} - \begin{bmatrix} 1 \\ \bar{\rho} \\ \rho \\ 1 \end{bmatrix} \begin{bmatrix} 1 & \rho & \bar{\rho} & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & a & b & 0 \\ 0 & b & a & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \qquad \begin{aligned} a &:= 1-|\rho|^2 \\ b &:= q-\rho^2 . \end{aligned} \end{aligned} \tag{5.4.1}$$

We immediately see that C_s has null space vectors

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \qquad \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} . \tag{5.4.2}$$

These are the desired null space vectors. The remaining 2×2 matrix in the center of (5.4.1) is hopefully nonsingular. If the sources are independent and circularly symmetric, then asymptotically (in n) $q \to 0$ and $\rho \to 0$, so that $a \to 1$ and $b \to 0$. Thus, for a sufficiently large number of samples it is clear that (with probability 1) the matrix is nonsingular. There is a danger for singularity if there exist specific relations between the two signals. In particular, if

$$\forall k: \quad s_1(k) = \pm\alpha s_2(k) \qquad (|\alpha| = 1)$$

(where the signs may be different for different k) then $\rho = \alpha|\rho|$, $q = \alpha^2$, $b = \alpha^2 a$, so that the center matrix is singular. This occurs in the case of two BPSK signals, for which $s(k) = \pm 1$,

or two MSK signals, for which $s(k) = \pm 1$, $s(k+1) = \pm j$, and also if two such signals have a constant phase offset or the same residual carrier modulation. Note that the phase offset α can be absorbed in the array matrix A, so that we can assume $\alpha = 1$. The additional null space vector in this case is

$$\begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix} . \tag{5.4.3}$$

The null space of C contains vectors $\mathbf{y}$ for which $[\bar{A} \otimes A]^* \mathbf{y}$ is a vector in the null space of C_s, i.e., either vector in (5.4.2), and perhaps the vector in (5.4.3). Assuming that A has full column rank, also $\bar{A} \otimes A$ has full column rank. Let $W = [\mathbf{w}_1 \ \mathbf{w}_2]$ be a separating beamformer such that $W^* A = I$, then

$$[\bar{A} \otimes A]^* [\bar{W} \otimes W] = \bar{A}^* \bar{W} \otimes A^* W = I \otimes I = I$$

from which we see that

$$[\bar{A} \otimes A]^* (\bar{\mathbf{w}}_1 \otimes \mathbf{w}_1) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} , \quad [\bar{A} \otimes A]^* (\bar{\mathbf{w}}_2 \otimes \mathbf{w}_2) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} ,$$

$$[\bar{A} \otimes A]^* (\bar{\mathbf{w}}_1 \otimes \mathbf{w}_2 - \bar{\mathbf{w}}_2 \otimes \mathbf{w}_1) = \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix} .$$

The solutions to $\mathbf{y}^* C \mathbf{y} = 0$ are thus spanned by a basis of the null space of $[\bar{A} \otimes A]^*$ (removed by prefiltering with dimension reduction) plus the desired solutions

$$\bar{\mathbf{w}}_1 \otimes \mathbf{w}_1 , \qquad \bar{\mathbf{w}}_2 \otimes \mathbf{w}_2 ,$$

plus, in the case of BPSK-type signals, an additional undesired solution

$$\bar{\mathbf{w}}_1 \otimes \mathbf{w}_2 - \bar{\mathbf{w}}_2 \otimes \mathbf{w}_1 .$$

If only the desired solutions are present in the null space of C, then the joint diagonalization step can find them from an arbitrary basis of this subspace. At this moment, it is not known how to deal with the undesired solution if it is present. Hence, we have to rule out cases with two or more BPSK signals and a general CM signal. However, for the case where all signals are BPSK, we we will propose a modified algorithm later in section 5.8.

The above analysis easily generalizes to more than two signals. A key property, valid for any number of signals and explicitly used by the algorithm, is the fact that certain columns (and rows) of C_s are identically zero. This property comes from $|s_k|^2 = 1$ alone and follows *by construction* for any number of samples. We do not have to wait for asymptotic convergence of the cross terms to zero. Many other blind source separation techniques require stochastic independence and rely on this. This aspect is the key to the good small-sample high-SNR performance of ACMA.

5.5 ACMA in noise

Let us now assume that our observations are noise perturbed: $\tilde{\mathbf{x}}_k = \mathbf{x}_k + \mathbf{n}_k$ $(k = 1, \cdots, n)$. Our objective in this section is to rederive the ACMA procedure by starting from the CMA(2,2) cost function

$$\mathbf{w} = \underset{\mathbf{w}}{\operatorname{argmin}}\ \mathrm{E}(|\hat{s}_k|^2 - 1)^2, \qquad \hat{s}_k = \mathbf{w}^* \tilde{\mathbf{x}}_k .$$

In a deterministic framework, a corresponding Least Squares problem can be introduced as

$$\mathbf{w} = \underset{\mathbf{w}}{\operatorname{argmin}}\ \frac{1}{n}\sum (|\hat{s}_k|^2 - 1)^2, \qquad \hat{s}_k = \mathbf{w}^* \tilde{\mathbf{x}}_k .$$

We will refer to this problem as the LS-CMA(2,2) problem.

Following the outline at the beginning of this section and using the same factorization as in (5.3.4), we can make a similar derivation:

$$\begin{aligned} \frac{1}{n}\sum (|\hat{s}_k|^2 - 1)^2 &= \frac{1}{n}\sum \left[(\bar{\tilde{\mathbf{x}}}_k \otimes \tilde{\mathbf{x}}_k)^* (\bar{\mathbf{w}} \otimes \mathbf{w}) - 1 \right]^2 \\ &= \frac{1}{n} \| \tilde{P}\mathbf{y} - \mathbf{1} \|^2 \qquad (\mathbf{y} = \bar{\mathbf{w}} \otimes \mathbf{w}) \\ &= \| \tilde{\mathbf{p}}^* \mathbf{y} - 1 \|^2 + \| \tilde{G}\mathbf{y} \|^2 . \end{aligned}$$

The next step is to replace the minimization of the first term by a fixed norm constraint on $\mathbf{y}$. It is well-known that this is possible in the usual situation where there is no structural constraint on $\mathbf{y}$. The following lemma shows that the structural constraint does not change this result.

Lemma 1: *Let*

$$\mathbf{x}_1 = \underset{\mathbf{x}=\bar{\mathbf{w}}\otimes\mathbf{w}}{\operatorname{argmin}} (\mathbf{a}^*\mathbf{x} - \beta)^2 + \| A\mathbf{x} \|^2, \qquad \textit{and} \quad \mathbf{x}_2 = \underset{\substack{\mathbf{x} = \bar{\mathbf{w}}\otimes\mathbf{w} \\ \mathbf{a}^*\mathbf{x} = \beta}}{\operatorname{argmin}} \| A\mathbf{x} \|^2 .$$

Then $\mathbf{x}_1$ *is proportional to* $\mathbf{x}_2$.

PROOF Define $\beta_1 = \mathbf{a}^*\mathbf{x}_1$. We can add the condition that $\mathbf{a}^*\mathbf{x} = \beta_1$ to the first optimization problem without changing the outcome:

$$\begin{aligned} \mathbf{x}_1 &= \underset{\substack{\mathbf{x} = \bar{\mathbf{w}}\otimes\mathbf{w} \\ \mathbf{a}^*\mathbf{x} = \beta_1}}{\operatorname{argmin}} (\mathbf{a}^*\mathbf{x} - \beta)^2 + \| A\mathbf{x} \|^2 \\ &= \underset{\substack{\mathbf{x} = \bar{\mathbf{w}}\otimes\mathbf{w} \\ \mathbf{a}^*\mathbf{x} = \beta_1}}{\operatorname{argmin}} (\beta_1 - \beta)^2 + \| A\mathbf{x} \|^2 = \underset{\substack{\mathbf{x} = \bar{\mathbf{w}}\otimes\mathbf{w} \\ \mathbf{a}^*\mathbf{x} = \beta_1}}{\operatorname{argmin}} \| A\mathbf{x} \|^2 . \end{aligned}$$

Scaling β_1 will scale the solution $\mathbf{x}_1$ accordingly, and does not affect the fact that it has a Kronecker structure. Hence

$$\mathbf{x}_1 \frac{\beta}{\beta_1} = \underset{\substack{\mathbf{x} = \bar{\mathbf{w}}\otimes\mathbf{w} \\ \mathbf{a}^*\mathbf{x} = \beta}}{\operatorname{argmin}} \| A\mathbf{x} \|^2 = \mathbf{x}_2 .$$

□

If we apply the lemma, it follows that the solution to the LS-CMA(2,2) problem is (up to a scaling) given by

$$\mathbf{w} = \underset{\substack{\mathbf{y}=\bar{\mathbf{w}}\otimes\mathbf{w} \\ \tilde{\mathbf{p}}^*\mathbf{y}=1}}{\operatorname{argmin}} \|\tilde{G}\mathbf{y}\|^2 .$$

The condition $\tilde{\mathbf{p}}^*\mathbf{y} = 1$ leads in a natural way to the choice of a prewhitening filter F as given in (5.2.12), viz.

$$\underline{\tilde{\mathbf{x}}} = F^*\tilde{\mathbf{x}}, \qquad \mathbf{w} = F\mathbf{t}, \qquad \text{where} \quad F = \hat{U}\hat{\Sigma}^{-1} .$$

Indeed, we derived before that

$$\tilde{\mathbf{p}}^*\mathbf{y} = \left(\frac{1}{n}\sum \bar{\tilde{\mathbf{x}}}\otimes\tilde{\mathbf{x}}\right)^* (\bar{\mathbf{w}}\otimes\mathbf{w}) = \mathbf{w}^*\tilde{R}_x\mathbf{w} .$$

If we change variables by prewhitening with dimension reduction, $\underline{\tilde{\mathbf{x}}} = \hat{\Sigma}^{-1}\hat{U}^*\tilde{\mathbf{x}}$ and $\mathbf{w} = \hat{U}\hat{\Sigma}^{-1}\mathbf{t}$, then $\underline{\tilde{R}}_x = I$ and

$$\mathbf{w}^*\tilde{R}_x\mathbf{w} = \mathbf{t}^*\mathbf{t} .$$

Moreover, $\|\underline{\mathbf{y}}\|^2 = \underline{\mathbf{y}}^*\underline{\mathbf{y}} = (\bar{\mathbf{t}}\otimes\mathbf{t})^*(\bar{\mathbf{t}}\otimes\mathbf{t}) = \bar{\mathbf{t}}^*\bar{\mathbf{t}}\otimes\mathbf{t}^*\mathbf{t} = \|\bar{\mathbf{t}}\|^2 \otimes \|\mathbf{t}\|^2 = \|\mathbf{t}\|^4$. It thus follows that the linear constraint on $\mathbf{y}$ can be replaced by a more pleasant unit-norm constraint on $\underline{\mathbf{y}}$ in the whitened domain:

$$\text{LS-CMA(2,2)} \quad \Leftrightarrow \quad \mathbf{w} = \underset{\substack{\mathbf{y}=\bar{\mathbf{w}}\otimes\mathbf{w} \\ \mathbf{w}^*\tilde{R}_x\mathbf{w}=1}}{\operatorname{argmin}} \|\tilde{G}\mathbf{y}\|^2 \quad \text{or} \quad \mathbf{t} = \underset{\substack{\mathbf{y}=\bar{\mathbf{t}}\otimes\mathbf{t} \\ \|\underline{\mathbf{y}}\|=1}}{\operatorname{argmin}} \|\underline{\tilde{G}}\underline{\mathbf{y}}\|^2 . \tag{5.5.1}$$

The first minimization problem is equivalent to the LS-CMA(2,2) problem up to a scaling which is not important. The second minimization problem is almost equal to the first, except that the whitening also involves a dimension reduction: this will force $\mathbf{w} = \hat{U}\hat{\Sigma}^{-1}\mathbf{t}$ to lie in the dominant column span of $\tilde{X}$.

At this point, ACMA and LS-CMA(2,2) will diverge in two distinct but closely related directions.

– LS-CMA(2,2) has to numerically optimize the first minimization problem in (5.5.1), and find d independent solutions. If we ignore the effect of the dimension reduction, the solutions will be unit-norm vectors $\underline{\mathbf{y}}$ that have the required Kronecker structure and minimize $\|\underline{\tilde{G}}\underline{\mathbf{y}}\|^2$. We expect that these solutions are close to the approximate nullspace of $\underline{\tilde{G}}$. Indeed, under noise-free conditions, they are precisely a basis of this null space.

– ACMA is making a twist on this problem: instead of solving for the true minimum, it first finds a basis for the the d-dimensional approximate nullspace of $\underline{\tilde{G}}$, then looks for unit-norm vectors in this subspace that best fit the required structure.

We thus see that ACMA and LS-CMA(2,2) are closely related provided we whiten the data using the noisy data covariance matrix $\tilde{R}_x$. The motivation for following the ACMA approach is that it is easier to solve for all d solutions (using joint diagonalization), whereas

the difference is likely to be not very large. Moreover, as we will show in section 5.6, the ACMA solution converges to the Wiener (LMMSE) solution, whereas CMA(2,2) is known to be unequal (but close) to the Wiener solution [57].

5.6 Asymptotic behavior

An analysis of the asymptotic behavior of ACMA in noise will reveal the close connections of this method with other blind source separation methods based on fourth-order moments.

In the noiseless case, we have derived in (5.3.5)

$$C := G^* G = \frac{1}{n}\sum(\overline{\mathbf{x}}_k \otimes \mathbf{x}_k)(\overline{\mathbf{x}}_k \otimes \mathbf{x}_k)^* - \frac{1}{n}\left[\sum \overline{\mathbf{x}}_k \otimes \mathbf{x}_k\right]\frac{1}{n}\left[\sum \overline{\mathbf{x}}_k \otimes \mathbf{x}_k\right]^* . \tag{5.6.1}$$

In the presence of noise, $\tilde{\mathbf{x}}_k = A\mathbf{s}_k + \mathbf{n}_k$, assume that we compute in the same way

$$\tilde{C} := \tilde{G}^* \tilde{G} = \frac{1}{n}\sum(\overline{\tilde{\mathbf{x}}}_k \otimes \tilde{\mathbf{x}}_k)(\overline{\tilde{\mathbf{x}}}_k \otimes \tilde{\mathbf{x}}_k)^* - \frac{1}{n}\left[\sum \overline{\tilde{\mathbf{x}}}_k \otimes \tilde{\mathbf{x}}_k\right]\frac{1}{n}\left[\sum \overline{\tilde{\mathbf{x}}}_k \otimes \tilde{\mathbf{x}}_k\right]^* \tag{5.6.2}$$

where $\tilde{G}$ is based on the noisy data. We analyze the contribution of the noise in this expression, assuming that it is zero mean, circularly symmetric, independent of the sources, and with finite covariance $R_n = \mathrm{E}(\mathbf{n}\mathbf{n}^*)$ and fourth-order moments.

Cumulants

The asymptotic analysis is best done via the introduction of fourth-order cumulants. For a zero-mean vector-signal $\mathbf{x}(k)$ with components $x_i(k)$, define the tensor with entries

$$\begin{aligned} \kappa_{i,k}^{j,l} &:= cum(x_i, \overline{x}_j, x_k, \overline{x}_l) \\ &:= \mathrm{E}(x_i \overline{x}_j x_k \overline{x}_l) - \mathrm{E}(x_i \overline{x}_j)\mathrm{E}(x_k \overline{x}_l) - \mathrm{E}(x_i \overline{x}_l)\mathrm{E}(x_k \overline{x}_j) - \mathrm{E}(x_i x_k)\mathrm{E}(\overline{x}_j \overline{x}_l) \end{aligned}$$

where $i, j, k, l = 1, \cdots, m$ and m is the dimension of $\mathbf{x}$. We will assume circularly symmetric sources (hence non-BPSK), so that the last term vanishes. If we collect the entries $\kappa_{i,k}^{j,l}$ into a matrix K_x with entries $K_{i+jm,l+km} = \kappa_{i,k}^{j,l}$, then

$$K_x = \mathrm{E}[(\overline{\mathbf{x}} \otimes \mathbf{x})(\overline{\mathbf{x}} \otimes \mathbf{x})^*] - \mathrm{E}[\overline{\mathbf{x}} \otimes \mathbf{x}]\mathrm{E}[\overline{\mathbf{x}} \otimes \mathbf{x})]^* - \mathrm{E}[\mathbf{x}\mathbf{x}^*]^T \otimes \mathrm{E}[\mathbf{x}\mathbf{x}^*]$$

Note that $\mathrm{E}[\mathbf{x}\mathbf{x}^*] = R_x$, $\mathrm{E}[\overline{\mathbf{x}} \otimes \mathbf{x}] = \mathrm{vec}(R_x)$. Comparing to (5.6.1), it is seen that, asymptotically,

$$C \doteq K_x + R_x^T \otimes R_x , \qquad \tilde{C} \doteq \tilde{K}_x + \tilde{R}_x^T \otimes \tilde{R}_x .$$

Cumulants are used because they have several well-known nice properties:

– *Multilinearity:* $\mathbf{x}_k = A\mathbf{s}_k$ implies

$$K_x = [\overline{A} \otimes A] K_s [\overline{A} \otimes A]^* .$$

– *Additivity:* Independent sources have additive cumulants in their sum. This has two important implications. If $\tilde{\mathbf{x}}_k = \mathbf{x}_k + \mathbf{n}_k$, where $\mathbf{n}$ is independent additive noise, then

$$\tilde{K}_x = K_x + K_n \,. \tag{5.6.3}$$

Let $\mathbf{s}_k = [s_1(k) \cdots s_d(k)]^T$ consist of independent signals and write $\mathbf{s}_k = \sum s_i(k)\mathbf{e}_i$, where $\mathbf{e}_i$ is the i-th unit coordinate vector. Let us also define the "auto-cumulants"

$$\kappa_i = cum(s_i, \bar{s_i}, s_i, \bar{s_i})$$

Then additivity implies

$$K_s = \sum_{i=1}^{d} \kappa_i [\mathbf{e}_i \otimes \mathbf{e}_i][\mathbf{e}_i \otimes \mathbf{e}_i]^T$$

In particular, K_s is a diagonal with only d nonzero entries κ_i.

– Gaussian sources have zero cumulants and hence disappear in (5.6.3): $K_n = 0$. Constant-modulus sources have auto-cumulant $\kappa_i = -1$.

For our model $\tilde{\mathbf{x}}_k = A\mathbf{s}_k + \mathbf{n}_k$, assuming independent circularly symmetric CM signals and independent Gaussian noise, this results in

$$\begin{aligned} \tilde{K}_x &= [\bar{A} \otimes A] K_s [\bar{A} \otimes A]^* + K_n \\ &= [\bar{A} \circ A] (-I) [\bar{A} \circ A]^* \end{aligned} \tag{5.6.4}$$

Note that in this asymptotic situation, the noise does not enter the equation.

Using these properties we can derive that, *without noise*, the CMA(2,2) or ACMA criterion matrix becomes asymptotically (using $R_x \doteq AA^*$)

$$\begin{aligned} C &\doteq K_x + R_x^T \otimes R_x \\ &\doteq [\bar{A} \otimes A] K_s [\bar{A} \otimes A]^* + \bar{A}\bar{A}^* \otimes AA^* \\ &= [\bar{A} \otimes A] (K_s + I) [\bar{A} \otimes A]^* \quad = \quad [\bar{A} \otimes A] C_s [\bar{A} \otimes A]^* \,. \end{aligned} \tag{5.6.5}$$

Note that $C_s = K_s + I$ is diagonal, with zero entries at the location of the source autocumulants, and '1' entries elsewhere on the diagonal. Like in the finite sample case, the null space of C_s is given by $\{\mathbf{e}_i \otimes \mathbf{e}_i\}$, and hence the null space of C by $\{\bar{\mathbf{w}}_i \otimes \mathbf{w}_i\}$, plus of course the null space of $[A \otimes A]^*$.

With noise, the CMA(2,2) or ACMA criterion matrix becomes asymptotically ($\tilde{R}_x \doteq R_x + R_n$)

$$\begin{aligned} \tilde{C} &\doteq \tilde{K}_x + \tilde{R}_x^T \otimes \tilde{R}_x \\ &\doteq K_x + (R_x + R_n)^T \otimes (R_x + R_n) + K_n \\ &= K_x + R_x^T \otimes R_x \quad + \quad R_x^T \otimes R_n + R_n^T \otimes R_x \quad + \quad K_n + R_n^T \otimes R_n \\ &= C \; + \; E \; + \; C_n \end{aligned} \tag{5.6.6}$$

where C is given in (5.6.5) and

$$E := R_x^T \otimes R_n + R_n^T \otimes R_x \,, \quad C_n := K_n + R_n^T \otimes R_n \,.$$

Thus, the noise contributes a second-order and a fourth-order term to the ACMA criterion matrix, even if it would be Gaussian. If we do not correct for it and proceed as in the noise-free case, this will result in a certain bias at the output of the beamformer. As we show next, this bias is precisely such that ACMA converges to the Wiener solution.

Asymptotic analysis of ACMA

In the analysis of ACMA, we also have to take the effect of the initial prewhitening step into account. Recall that this step is $\tilde{\underline{X}} = F^* \tilde{X} = \underline{A}S + \underline{N}$, where $F = \hat{U}\hat{\Sigma}^{-1}$ so that $\tilde{\underline{R}}_x = F^* \tilde{R}_x F = I$. In the whitened domain, we search for d-dimensional beamformers $\mathbf{t}$, the overall beamformers are then given by $\mathbf{w} = F\mathbf{t}$.

If we redo the derivation in (5.6.6), but in a different direction and assuming Gaussian noise ($K_n = 0$), we obtain that asymptotically

$$\begin{aligned} \tilde{\underline{C}} &= (F \otimes F)^* \tilde{C} (F \otimes F) \\ &\doteq (F \otimes F)^* K_x (F \otimes F) + I \otimes I \\ &= [\bar{\underline{A}} \otimes \underline{A}] K_s [\bar{\underline{A}} \otimes \underline{A}]^* + I \\ &= [\bar{\underline{A}} \circ \underline{A}](-I)[\bar{\underline{A}} \circ \underline{A}]^* + I . \end{aligned} \tag{5.6.7}$$

Inserting this in the CMA(2,2) cost function, equation (5.5.1), it follows that both ACMA and CMA(2,2) look at the optimization problem

$$\begin{aligned} \underset{\mathbf{y}=\bar{\mathbf{t}}\otimes\mathbf{t}, \|\mathbf{y}\|=1}{\operatorname{argmin}} \; \mathbf{y}^* \tilde{\underline{C}} \mathbf{y} &\doteq \underset{\mathbf{y}=\bar{\mathbf{t}}\otimes\mathbf{t}, \|\mathbf{y}\|=1}{\operatorname{argmin}} \; \mathbf{y}^* \left\{ [\bar{\underline{A}} \circ \underline{A}](-I)[\bar{\underline{A}} \circ \underline{A}]^* + I \right\} \mathbf{y} \\ &= \underset{\mathbf{y}=\bar{\mathbf{t}}\otimes\mathbf{t}, \|\mathbf{y}\|=1}{\operatorname{argmax}} \; \mathbf{y}^* \left\{ [\bar{\underline{A}} \circ \underline{A}][\bar{\underline{A}} \circ \underline{A}]^* \right\} \mathbf{y} . \end{aligned} \tag{5.6.8}$$

CMA(2,2) continues to optimize this problem. As we motivate below, the result is in general *not* the desired vectors of the form $\bar{\underline{\mathbf{a}}}_i \otimes \underline{\mathbf{a}}_i$. ACMA is taking a slightly different approach at this point: it does not optimize (5.6.8), but solves the unstructured problem first. Indeed, it looks for an unconstrained d-dimensional basis $\{\mathbf{y}_i\}$ of the approximate null space of $\tilde{\underline{G}}$, equivalently the null space of $\tilde{\underline{C}}$, or d independent unit-norm vectors $\mathbf{y}$ that minimize $\mathbf{y}^* \tilde{\underline{C}} \mathbf{y}$. With the factorization in (5.6.7), we see that these are the d dominant eigenvectors of $[\bar{\underline{A}} \circ \underline{A}][\bar{\underline{A}} \circ \underline{A}]^*$. Since this is a rank-$d$ matrix, we have that the d dominant eigenvectors together span the same subspace as the column span of $[\bar{\underline{A}} \circ \underline{A}]$, hence

$$\operatorname{span}\{\mathbf{y}_1, \cdots, \mathbf{y}_d\} \doteq \operatorname{span}[\bar{\underline{A}} \circ \underline{A}] = \operatorname{span}\{\bar{\underline{\mathbf{a}}}_1 \otimes \underline{\mathbf{a}}_1, \cdots, \bar{\underline{\mathbf{a}}}_d \otimes \underline{\mathbf{a}}_d\} .$$

As a second step, it will use the joint diagonalization to replace the unstructured basis by one that has the required Kronecker product structure, i.e., look for d independent vectors of the form $\bar{\mathbf{t}} \otimes \mathbf{t}$ within this column span. From the above equation, we see that the unique solution is

$$\bar{\mathbf{t}}_i \otimes \mathbf{t}_i = \bar{\underline{\mathbf{a}}}_i \otimes \underline{\mathbf{a}}_i , \qquad i = 1, \cdots, d$$

(up to a scaling to make $\mathbf{t}_i$ have unit norm) and thus

$$\mathbf{t}_i = \underline{\mathbf{a}}_i , \qquad i = 1, \cdots, d .$$

The beamformer on the whitened problem is equal to the whitened direction vector (a matched spatial filter). If we go back to the resulting beamformer on the original (unwhitened) data matrix X, we find

$$\mathbf{t}_i = \underline{\mathbf{a}}_i = F^* \mathbf{a}_i \qquad \Rightarrow \qquad \mathbf{w}_i = F\mathbf{t}_i = FF^*\mathbf{a}_i = \tilde{R}_x^{-1}\mathbf{a}_i\,, \qquad (i = 1, \cdots, d) \tag{5.6.9}$$

since $F = \hat{U}\hat{\Sigma}^{-1}$, $\tilde{R}_x = U\Sigma^2 U^* \doteq \hat{U}\hat{\Sigma}^2\hat{U} + \sigma^2\hat{U}^{\perp}\hat{U}^{\perp *}$, and $\hat{U}^{\perp *}\mathbf{a}_i \doteq 0$. We have just shown that ACMA is asymptotically equal to the Wiener receiver (or LMMSE beamformer). In general, this is a very attractive property.

Does this two-step procedure solve the CMA(2,2) optimization problem (5.6.8)? This is not likely, since in this asymptotic case ACMA finds its structured solutions only inside the subspace spanned by the columns of $[\underline{\bar{A}} \circ \underline{A}]$. A solution to CMA(2,2) is expected to be close to a dominant eigenvector of $[\underline{\bar{A}} \circ \underline{A}][\underline{\bar{A}} \circ \underline{A}]^*$, but it is not restricted to be inside the subspace. Thus, if the eigenvectors are not equal to $\{\underline{\bar{\mathbf{a}}}_i \otimes \underline{\mathbf{a}}_i\}$, the CMA(2,2) optimal solution might be different. This happens if the columns of $\underline{A}$ are not orthogonal. But there are only two situations where the columns of $\underline{A}$ are precisely orthogonal: if there is no noise, or (assuming white Gaussian noise) if the columns of the unwhitened A are orthogonal. This is a rather special case, approximately true if the sources are well separated and the number of sensors is large. Thus, CMA(2,2) does in general not lead to the Wiener solution. This result matches that in the equalization context [21], see also chapter 8 in volume 1.

Connection to JADE

A widely used algorithm for the blind separation of independent non-Gaussian sources in Gaussian noise is JADE ("Joint Approximate Diagonalization of Eigen-matrices") [11]. It is based on the construction of the fourth-order cumulant matrix $\tilde{K}_x$ in equation (5.6.4), but uses the alternative prefiltering strategy as in (5.2.13), i.e., $F = \hat{U}(\hat{\Sigma}^2 - \sigma^2 I)^{-1/2}$ where $\hat{U}$ and $\hat{\Sigma}$ are estimated from the eigenvalue decomposition of $\tilde{R}_x$. The prefiltering leads to $\underline{\underline{\tilde{X}}} = F^*\tilde{X} = \underline{\underline{A}}S + \underline{\underline{N}}$, where $\underline{\underline{A}} = F^*A$. This choice is motivated by the fact that, asymptotically, F converges to $F \doteq U_A\Sigma_A^{-1}$ (based on the SVD $A = U_A\Sigma_A V_A$), and thus

$$\underline{\underline{A}} \doteq \Sigma_A^{-1}U_A^*A = V_A$$

is a unitary matrix. Asymptotically, the fourth-order cumulant matrix is given by

$$\underline{\underline{\tilde{K}}}_x \doteq [\underline{\underline{\bar{A}}} \circ \underline{\underline{A}}]\,(-I)\,[\underline{\underline{\bar{A}}} \circ \underline{\underline{A}}]^*\,.$$

JADE computes a basis of the dominant column span of this matrix, which in this asymptotic situation spans the same subspace as

$$\{\underline{\underline{\bar{\mathbf{a}}}}_i \otimes \underline{\underline{\mathbf{a}}}_i\,;\; i = 1, \cdots, d\}$$

Like ACMA, it then performs a joint diagonalization to identify the vectors $\underline{\underline{\mathbf{a}}}_i$. After correcting for the prefiltering, we find

$$T = \underline{\underline{A}} = V_A \quad \Rightarrow \quad W = FT = U_A\Sigma_A^{-1}V_A = A^{\dagger *}\,.$$

Hence, this strategy leads asymptotically to the zero-forcing beamformer, as well as the true A-matrix.

Apart from different prefiltering, the asymptotic equations of JADE and ACMA look rather similar. JADE searches for eigenvectors corresponding to nonzero eigenvalues given by the nonzero entries of K_s, here equal to -1, whereas ACMA looks for the null space vecors generated by the zero entries of $K_s + I$. The result is the same.

However, the finite-sample properties are quite different. In the absence of noise, the null space information of $\tilde{C}$ in ACMA is exact by construction, and hence the algorithm produces the exact separating beamformers. The dominant column span of $\tilde{K}$ used in JADE is not exact since the signal sources do not decorrelate exactly in finite samples: K_s is a full matrix. Thus, keeping the number of samples fixed, the SNR-asymptotic performance of JADE saturates.

In the proposed implementation in [11], JADE explicitly uses the fact that (with the $\Sigma_A^{-1}U_A^*$-prefiltering), $\underline{\underline{A}} = V_A$ and hence unitary. It thus forces the joint diagonalization to produce a unitary matrix. A finite-sample problem is that $\tilde{R}_x$ does not reveal yet the true U_A and Σ_A, and the restriction might make the results less accurate. This problem was noted in [9], where optimal combinations of second and fourth order statistics are presented.

In summary, we can say that JADE and ACMA are quite similar, but differ in the following points:

– Prefiltering scheme, so that ACMA converges to a Wiener solution and JADE to a zero-forcing beamformer,

– JADE explicitly relies on stochastic independence of sources, whereas ACMA explicitly relies on the CM property. This leads to different finite sample behavior.

To finish this section, we mention that the problem considered by JADE and the related ICA [13] may be formulated as a least squares problem

$$\min_{V,K_s} \| \underline{\underline{\tilde{K}}}_x - [\overline{V} \circ V] K_s [\overline{V} \circ V]^* \|_F^2 \tag{5.6.10}$$

where $V = V_A$ is unitary, and K_s is a diagonal matrix containing the (unknown) source cumulants. JADE gives an approximate but algebraically computable solution to this problem [56]. The true solution to the minimization problem is considered in [55]; it is shown to be equivalent to the ICA criterion, with somewhat better performance than JADE but computationally less attractive. An algorithm for solving (5.6.10) was proposed in [39].

5.7 Weighted ACMA

We have seen before, in (5.6.6), that asymptotically

$$\tilde{C} \doteq C + E + C_n\,, \qquad \text{where } E := R_x^T \otimes R_n + R_n^T \otimes R_x\,, \quad C_n := K_n + R_n^T \otimes R_n\,.$$

Thus $\tilde{C}$ is in expectation equal to the noise-free C, plus a second-order and fourth-order contribution E and C_n due to noise. E and C_n represent bias terms that cause ACMA to converge to the Wiener solution. If an unbiased estimate of A is desired (e.g., for direction estimation),

then the bias has to be removed. Although the Wiener beamformer can easily be mapped to a zero-forcing beamformer (by multiplication by $\tilde{R}_x$), the finite-sample properties of this are not so good. Here we look at an alternative, based on correcting $\tilde{C}$.

Let us assume that we know the noise covariance R_n. We cannot know E since it depends on noise-free data, but we can construct

$$\tilde{E} \;:=\; (\tfrac{1}{n}\sum \tilde{\mathbf{x}}\tilde{\mathbf{x}}^*)^T \otimes R_n + R_n^T \otimes (\tfrac{1}{n}\sum \tilde{\mathbf{x}}\tilde{\mathbf{x}}^*) \quad = \quad \tilde{R}_x^T \otimes R_n + R_n^T \otimes \tilde{R}_x\,. \tag{5.7.1}$$

Asymptotically,

$$\tilde{E} \quad \doteq \quad R_x^T \otimes R_n + R_n^T \otimes R_x \;+\; 2R_n^T \otimes R_n$$

so that

$$\tilde{C} \;-\; \tilde{E} \;-\; (K_n - R_n^T \otimes R_n) \quad \doteq \quad C$$

is an asymptotically unbiased estimate of C. If the noise is Gaussian, then $K_n = 0$. If also we can assume that $\|R_n\|_F^2 \ll \|R_x\|_F^2$, i.e., the SNR is sufficiently large, then we can ignore $R_n^T \otimes R_n$ compared to $\tilde{E}$ as well, and use $\tilde{C} - \tilde{E}$ to estimate C.

Let us now assume that we know the noise covariance only up to a scalar, i.e., suppose that $\mathrm{E}(\mathbf{n}\mathbf{n}^*) = \sigma^2 R_n$ where σ is unknown. Computing $\tilde{E}$ as in (5.7.1), it follows that we have available the data matrices $\tilde{C}$ and $\tilde{E}$, satisfying the approximate model (ignoring 4-th order terms)

$$\tilde{C} \;\simeq\; C + \sigma^2 \tilde{E}\,.$$

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

$$(\tilde{C} - \lambda \tilde{E})\mathbf{y} \;=\; \mathbf{0}\,.$$

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

Alternatively, recalling the factorization $\tilde{C} = \tilde{G}^* \tilde{G}$, we can use $B := \tilde{E}^{1/2}$ to prewhiten the data $\tilde{G}$:

$$\begin{array}{lcl} (\tilde{C} - \lambda \tilde{E})\mathbf{y} = \mathbf{0} & \Leftrightarrow & B(B^{-1}\tilde{C}B^{-1} - \lambda I)B\mathbf{y} = \mathbf{0} \\ & \Leftrightarrow & (\tilde{G}'^* \tilde{G}' - \lambda I)\mathbf{y}' = \mathbf{0}\,, \end{array}$$

where

$$\begin{array}{rcl} \tilde{G}' & := & \tilde{G}B^{-1} \\ \mathbf{y}' & := & B\mathbf{y}\,. \end{array}$$

Thus we compute $\{\mathbf{y}_i'\}$ as the d least significant right singular vectors of $\tilde{G}B^{-1}$, and then set $\mathbf{y}_i = B^{-1}\mathbf{y}_i'$. At this point, we can continue with the joint diagonalization and recover the beamforming matrix W. Asymptotically in n and SNR, we obtain $W \doteq A^{\dagger *}$.

The algorithm is called Weighted-ACMA (WACMA). As in ACMA, a dimension-reducing prefiltering F is necessary. If we take the same prewhitening prefilter as in ACMA, then after whitening, $\underline{\tilde{R}}_x = I$ and $\underline{R}_n = \hat{\Sigma}^{-2}$. Thus, $\tilde{E} = I \otimes \hat{\Sigma}^{-2} + \hat{\Sigma}^{-2} \otimes I$ is diagonal, and is easily factored: $B = E^{1/2}$ is diagonal. The resulting algorithm is summarized in figure 5.5. Simulations comparing ACMA, WACMA and JADE appear in section 5.9.

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

1. SVD: $R_n^{-1/2}\tilde{X} =: U\Sigma V$
   Prewhitening/dimension reduction: $\underline{\tilde{X}} := \hat{\Sigma}^{-1}\hat{U}^* R_n^{-1/2}\tilde{X}$
   Construct P with rows $\mathrm{vech}(\underline{\tilde{\mathbf{x}}}_k\underline{\tilde{\mathbf{x}}}_k^*)^T$
   QR fact.: $Q[\mathbf{1} \quad P] =: \sqrt{n}\begin{bmatrix} 1 & \mathbf{p}^* \\ \mathbf{0} & G \end{bmatrix}$
   $E := I \otimes \hat{\Sigma}^{-2} + \hat{\Sigma}^{-2} \otimes I$
   SVD: $\{\mathbf{y}_i'\} = \ker(GE^{-1/2})$
   $\mathbf{y}_i = E^{-1/2}\mathbf{y}_i' \ \ (i = 1, \cdots, d)$
   $Y_i = \mathrm{vech}^{-1}\mathbf{y}_i \ \ (i = 1, \cdots, d)$
2. Continue as in the usual ACMA, step 2 (figure 5.4)

Figure 5.5. Weighted ACMA

5.8 Binary source separation

The blind *binary* souce separation problem is to find a factorization $X = AS$ where $S_{ij} \in \{\pm 1\}$. It is thus a specialization of our previous model, where in addition S is real-valued. The latter property allows us to write

$$X = AS \quad \Leftrightarrow \quad \begin{bmatrix} \mathrm{real}(X) \\ \mathrm{imag}(X) \end{bmatrix} = \begin{bmatrix} \mathrm{real}(A) \\ \mathrm{imag}(A) \end{bmatrix} S \quad \Leftrightarrow \quad X_R = A_R S\,,$$

with $X_R \in \mathbb{R}^{2m\times n}$ and $A_R \in \mathbb{R}^{2m\times d}$. This forces S to be real, and at the same time, A_R is usually much better conditioned than A, thus improving its robustness to noise. The problem is to find all independent vectors $\mathbf{w}_R \in \mathbb{R}^{2M}$ such that $\mathbf{w}_R^T X_R = \mathbf{s}$ has entries $s_k \in \{\pm 1\}$.

The alphabet condition is written as

$$s_k \in \{\pm 1\} \quad \Leftrightarrow \quad (s_k - 1)(s_k + 1) = 0 \quad \Leftrightarrow \quad s_k^2 = 1 \tag{5.8.1}$$

(with possible extensions to other constellations). Denoting the k-th column of X_R by $\mathbf{x}_k$, substitution of $\mathbf{w}_R^T\mathbf{x}_k = s_k$ into (5.8.1) leads to

$$\begin{aligned} \mathbf{w}^T\mathbf{x}_k\mathbf{x}_k^T\mathbf{w} &= 1\,, \qquad k = 1, \cdots, n\,. \\ \Leftrightarrow \quad [\mathbf{x}_k \otimes \mathbf{x}_k]^T[\mathbf{w}_R \otimes \mathbf{w}_R] &= 1\,. \end{aligned} \tag{5.8.2}$$

If we now continue as in ACMA, we arrive at the problem $P\mathbf{y} = \mathbf{1}$, where $P = X_R \circ X_R$ and $\mathbf{y} = \mathbf{w}_R \otimes \mathbf{w}_R$. However, this vector $\mathbf{y}$ has many duplicate entries: it corresponds to a matrix $Y = \mathbf{w}\mathbf{w}^T$ which is real symmetric. In the same way, it follows that the corresponding columns of P are repeated. This will give rise to additional undesired null space solutions, causing the joint diagonalization step to fail.

It is in the present situation rather straightforward to remove the duplicate entries. Define, for a $d \times d$ real symmetric matrix $Y = [y_{ij}]$, a scaled stacking of the lower triangular part of the columns:

$$\text{vecr}(Y) := [y_{11} \quad y_{21}\sqrt{2} \cdots y_{d1}\sqrt{2} \quad y_{22} \quad y_{32}\sqrt{2} \cdots y_{d,d-1}\sqrt{2} \quad y_{dd}]^T \in \mathbb{R}^{\,d(d+1)/2}.$$

Corresponding to this linear operation, there exists an orthogonal matrix J_R of size $d(d+1)/2 \times d^2$ and with a simple structure such that

$$J_R(\mathbf{w}_R \otimes \mathbf{w}_R) = \text{vecr}(\mathbf{w}_R \mathbf{w}_R^T)$$

Since $J_R^T J_R = I$, we can write (5.8.2) as

$$[\mathbf{x}_k \otimes \mathbf{x}_k]^T [\mathbf{w}_R \otimes \mathbf{w}_R] \;=\; [\mathbf{x}_k \otimes \mathbf{x}_k]^T J_R^T \cdot J_R [\mathbf{w}_R \otimes \mathbf{w}_R] \;=\; 1 \qquad k = 1, \cdots, n. \tag{5.8.3}$$

After collecting all rows $[\mathbf{x}_k \otimes \mathbf{x}_k]^T J_R^T = [\text{vecr}(\mathbf{x}_k \mathbf{x}_k^T]^T$ into a matrix P_R, the problem reduces to finding all independent vectors $\mathbf{y}_R$ satisfying

$$P_R \mathbf{y}_R = \mathbf{1}, \qquad \mathbf{y}_R = J_R(\mathbf{w}_R \otimes \mathbf{w}_R)$$

We can now follow the same procedure as in ACMA, construct G_R from P_R, and find a d-dimensional basis $\{(\mathbf{y}_R)_i\}$ of its null space. The structural property $\mathbf{y}_R = J_R(\mathbf{w}_R \otimes \mathbf{w}_R)$ implies $J_R^T \mathbf{y}_R = \mathbf{w}_R \otimes \mathbf{w}_R$. We can thus write $Y_i = \text{vecr}^{-1}(\mathbf{y}_R)_i = \text{vec}^{-1}(J_R^T \mathbf{y}_R)$, which gives

$$\mathbf{w}\mathbf{w}^T = \alpha_1 Y_1 + \cdots + \alpha_\delta Y_\delta \,. \tag{5.8.4}$$

The rest is the same as in ACMA, except that all matrices are real. The algorithm is called RACMA [46].

Other algorithms Other algorithms to solve the binary source separation problem (or separation of sources with more extended alphabets) are ILSE and ILSP [43], which are similar to the first algorithm in (5.2.9), ILSF [52] and DWILSP [37], which are related to the second algorithm in (5.2.9) and (5.1.3), respectively. As with other alternating projections algorithms, the main concern with these algorithms is their initialization and lack of global convergence. Depending on the initialization, the algorithms can converge to a local minimum, and restarts are needed if not all independent signals are found. A maximum-likelihood technique based on the EM algorithm was proposed in [6], and another one later in [24]. These are iterative algorithms that require an accurate initialization such as provided by RACMA or JADE.

Several people noted that the factorization problem $X = AS$ is essentially a clustering problem: in the absence of noise, X can contain only 2^d distinct vectors. To estimate A, it suffices to determine a suitable assignment of these vectors (or cluster centers) to constellation vectors, i.e., the columns of S, taking symmetry into account [3], [4], [25]. With noise, however, the segmentation and hence the estimation of the cluster centers can be difficult and limits the performance of such algorithms.

Simulations comparing several of these algorithms to RACMA have appeared in [46]. As an application of this algorithm, we mention the blind separation and equalization of GSM signals [51].

5.9 Simulations

Some performance results are shown in figures 5.6 and 5.7. In this simulation, we took a ULA($\frac{\lambda}{2}$) consisting of $m = 4$ antennas, and $d = 3$ equal-power constant-modulus sources with directions $-10°, 0°, 20°$ respectively. We compare the performance of ACMA, W-ACMA and JADE.

In figure 5.6, 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. The reference performance is that of a zero-forcing (ZF) beamformer based on sample data ($\hat{W} = \hat{A}^{\dagger *}$, $\hat{A} = \tilde{X}S^{\dagger}$, assuming known S). Figure 5.7 shows the same but for the Signal to Interference and Noise ratio (SINR). Here, the reference performance is that of a Wiener receiver based on sample data ($\hat{W} = S\tilde{X}^{\dagger}$ with known S).

Figure 5.6 (right column) shows that the SIR performance of JADE saturates for finite n because it relies on the convergence of fourth-order statistics, whereas (left column) the SIR performance of ACMA saturates for finite SNR, because it converges to the Wiener solution and hence it is biased. It is seen that the whitening in W-ACMA removes this saturation so that it can converge to a few dB below the ZF solution. Figure 5.7 shows that ACMA converges asymptotically (in n) to the Wiener solution.

Finally, figure 5.8 shows the SIR and SINR for three equal-powered sources with directions $[-\alpha, 0, \alpha]$, for varying α. The SNR was 10 dB, and we took $n = 200$ samples. For large α, the columns of A become approximately orthogonal, $\hat{\Sigma} \approx I$, and the difference between ACMA and WACMA disappears. The performance of JADE is limited by the finite sample effect.

5.10 Joint diagonalization

In the preceding sections, we have mostly analyzed the first step of the algorithm: up to the point of finding a basis $\{\mathbf{y}_i\}$ of the null space of $\underline{\tilde{G}}$. We finish the chapter by going into more details on the second step: how to identify from this basis the structured beamforming vectors $\{\bar{\mathbf{t}}_i \otimes \mathbf{t}_i\}$ spanning the same space. The solution is given by a joint diagonalization of the matrices Y_i constructed from $\mathbf{y}_i$.

Finding the rank-one basis

After we have found a basis $\{\mathbf{y}_i\}$ for the (approximate) null space of $\underline{\tilde{G}}$, we have to find which basis $\{\bar{\mathbf{t}}_i \otimes \mathbf{t}_i\}$ spans the same subspace. By rearranging the d^2-dimensional vectors as $d \times d$ matrices, as in (5.3.2), we have seen that this problem can be written as

$$\begin{array}{lcll} Y_1 & = & T\Lambda_1 T^* & (\Lambda_i \text{ diagonal}) \\ Y_2 & = & T\Lambda_2 T^* & \\ & \vdots & & \\ Y_d & = & T\Lambda_d T^*\,. & \end{array} \tag{5.10.1}$$

This problem is known as a joint (or simultaneous) diagonalization problem, by congruence. It is related to generalized eigenvalue problems, which is seen by postmultiplying by Y_1^{-1}

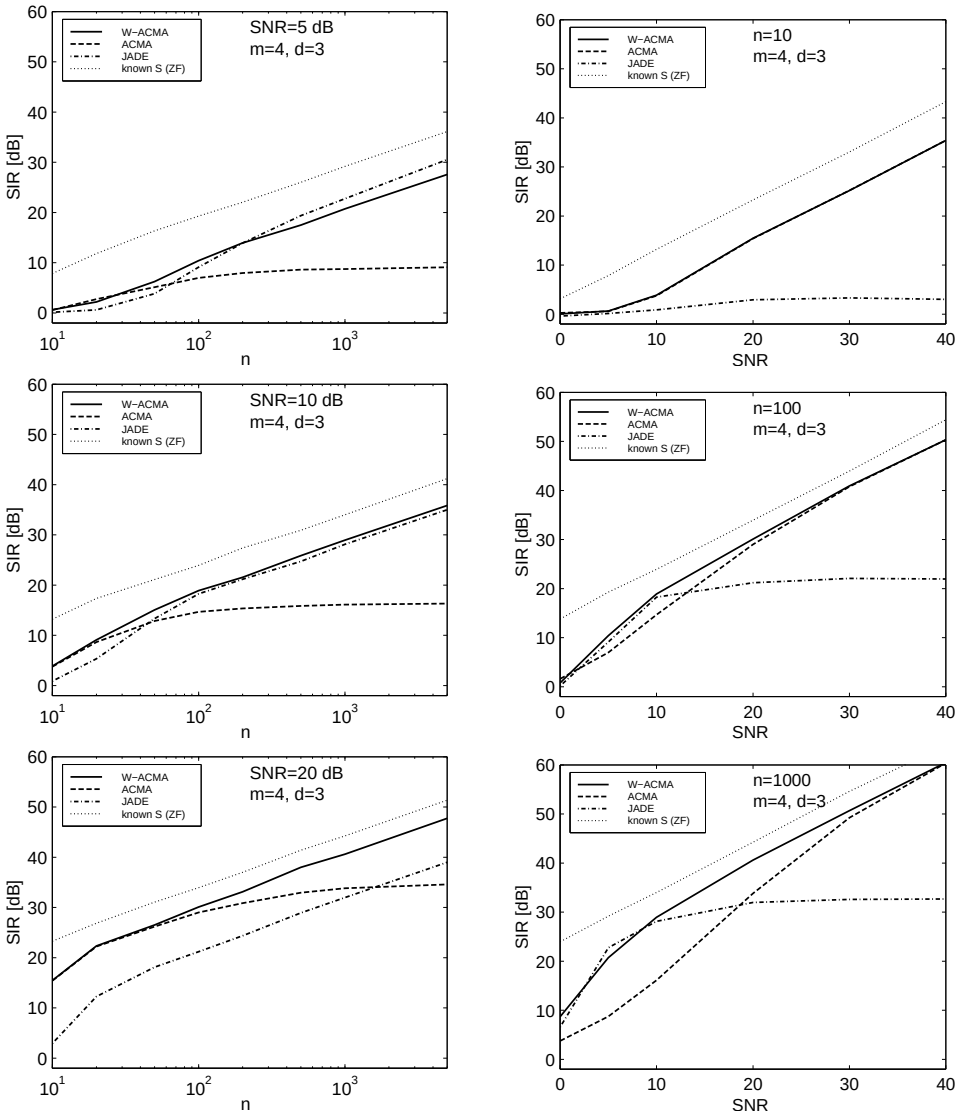

Figure 5.6. SIR performance of W-ACMA, ACMA and JADE, as function of *n* and SNR

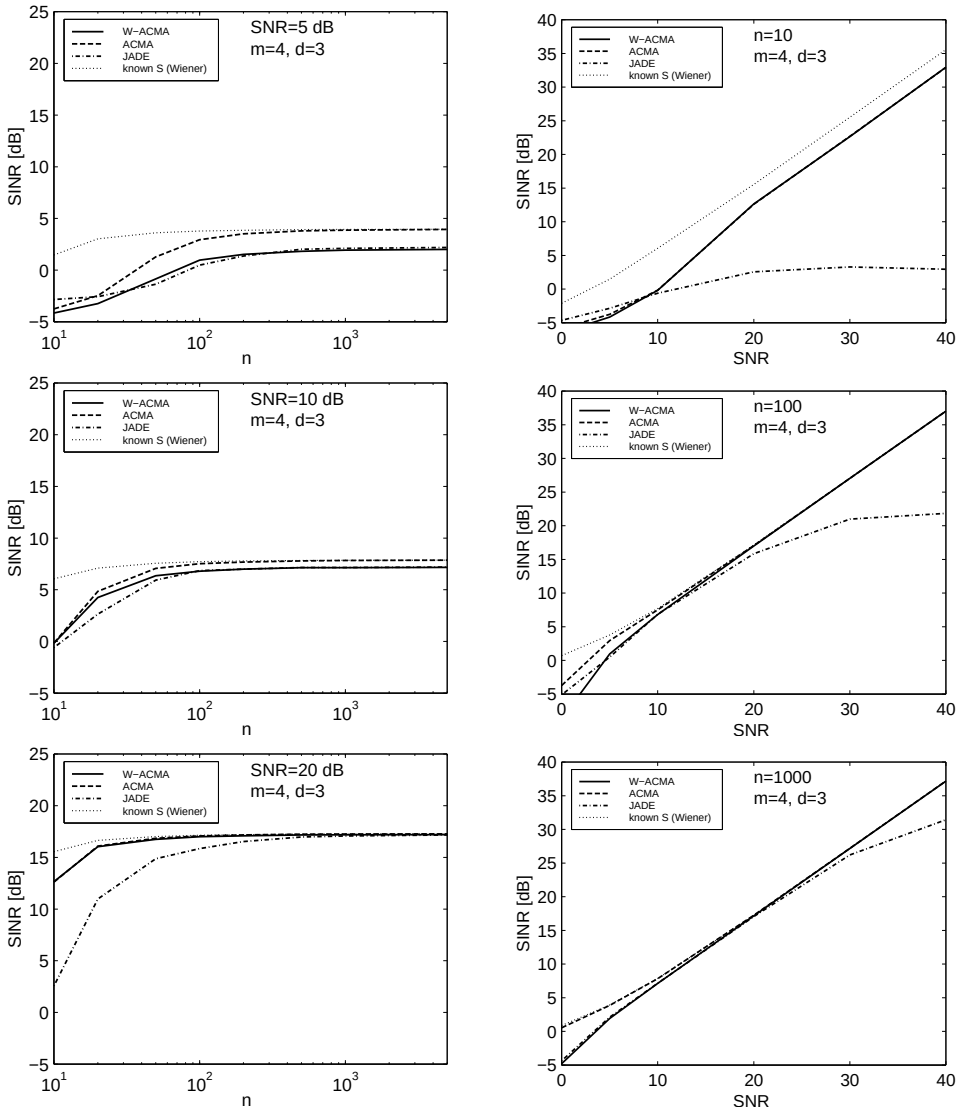

Figure 5.7. SINR performance of W-ACMA, ACMA and JADE

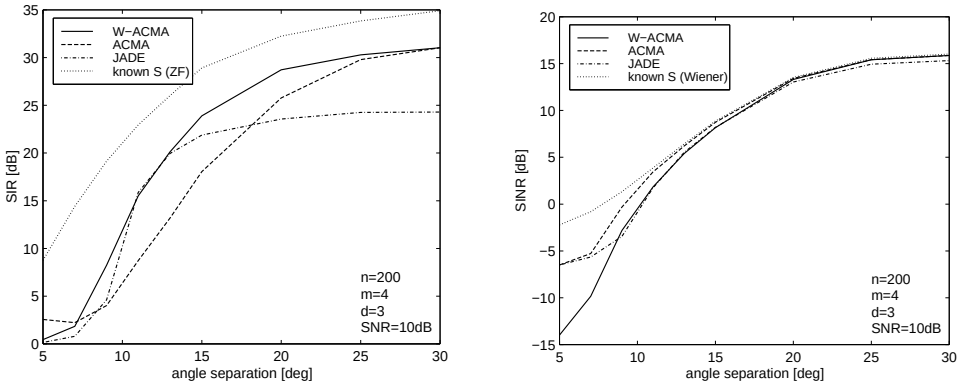

Figure 5.8. SIR and SINR performance for varying angle separation

(assuming it is invertible), so that we obtain $Y_i Y_1^{-1} = T(\Lambda_i \Lambda_1^{-1}) T^{-1}$. If the eigenvalues of one such product are unique, then T is unique up to scaling and permutation, and we can solve the problem. With $i > 2$, the problem is in general overdetermined, with no exact solution in the noisy case. Numerically it is better to avoid the inversion and to take all matrices Y_i into account. One approach for this is given below; a literature overview is given at the end of the section.

Bring in a QR factorization of T^* and an RQ decomposition of T:

$$T = QR', \qquad T^* = R''Z$$

where Q, Z are unitary, and R', R'' are upper triangular and invertible. The factorizations are of course related, but we will ignore this for the moment. Substitution into (5.10.1) leads to

$$\begin{array}{rcl} Q^* Y_1 Z & = & R_1 \\ Q^* Y_2 Z & = & R_2 \\ & \vdots & \\ Q^* Y_d Z & = & R_d \end{array} \tag{5.10.2}$$

with $R_1, \cdots, R_d$ upper triangular:

$$\begin{array}{rcl} R_1 & = & R' \Lambda_1 R'' \\ R_2 & = & R' \Lambda_2 R'' \\ & \vdots & \\ R_d & = & R' \Lambda_d R''. \end{array} \tag{5.10.3}$$

Hence, there exists Q, Z such that all $Q^* Y_k Z$ are upper triangular. This is a generalized Schur decomposition, but for d matrices rather than two.

Suppose that we have found this decomposition, then how do we reconstruct T? From the fact that $\{\mathbf{y}_i\}$ and $\{\bar{\mathbf{t}}_i \otimes \mathbf{t}_i\}$ are both bases for the same subspace, we can formulate that

there exist linear combinations of the Y_j such that the result is $\mathbf{t}_i\mathbf{t}_i^*$. This implies that there are complex numbers α_{ij} such that

$$\sum_j \alpha_{ij} Y_j \;=\; \mathbf{t}_i\mathbf{t}_i^*$$

In particular, the result is rank 1. Introducing the QZ factorizations (5.10.2), it follows that

$$\sum_j \alpha_{ij} R_j\,, \qquad i = 1, \cdots, d$$

is also rank 1. But since this matrix is upper triangular as well, at most one diagonal entry of the result can be nonzero. (From the structure of the R_i in (5.10.3) and the invertibility of R', R'', we deduce that *precisely* one diagonal entry is nonzero.) Thus, it suffices to look at linear combinations of the diagonals of the R_j. Collect these diagonals in a matrix R,

$$\mathbf{r}_j := \mathrm{diag}[R_j]\,, \qquad R := [\mathbf{r}_1 \quad \mathbf{r}_2 \;\cdots\; \mathbf{r}_d]$$

and let A be a matrix with entries α_{ij}. Setting for the i-th linear combination the i-th diagonal entry of the result equal to 1, we find

$$\sum_j \alpha_{ij}\mathbf{r}_j = \mathbf{e}_i\,, \quad (i = 1, \cdots, d) \qquad \Leftrightarrow \qquad AR = I \qquad \Leftrightarrow \qquad A = R^{-1}\,.$$

Having found the coefficients $\{\alpha_{ij}\}$, we can now compute d linear combinations $\sum_j \alpha_{ij}Y_j$ $(i = 1, \cdots, d)$ and factor the result as $\mathbf{t}_i\mathbf{t}_i^*$. The $\mathbf{t}_i$ form the columns of the matrix T. Hence, in the noise-free case, the computation of a generalized Schur decomposition (5.10.2) gives the solution to the simultaneous diagonalization problem.

Generalized Schur decomposition

Let us now see how we can compute the generalized Schur decomposition problem (5.10.2) in the presence of noise. Since the problem is overdetermined for $d > 2$, in general there will be no Q, Z that will make all $R_1, \cdots, R_d$ upper triangular, but we can try to find a best fit:

$$\begin{aligned} \min_{T,\{\Lambda_k\}} \sum_{k=1}^d \|\, Y_k - T\Lambda_k T^* \,\|_F^2 \quad &\approx \quad \min_{Q,Z,\{R_k\}} \sum_{k=1}^d \|\, Y_k - QR_kZ^* \,\|_F^2 \\ &= \quad \min_{Q,Z,\{R_k\}} \sum_{k=1}^d \|\, Q^*Y_kZ - R_k \,\|_F^2\,. \end{aligned}$$

It follows that for any Q, Z the best fit for R_k is equal to the upper triangular part of Q^*Y_kZ, so that the modeling error becomes equal to the norm of the strictly lower triangular part. Thus, we have to make all Q^*Y_kZ as much upper triangular as possible, or minimize the strictly lower triangular entries. Our approach is to modify the standard QZ or Jacobi iteration method used for computing the Schur decomposition of two matrices [34], [18] so that it works for more than two matrices.

The QZ iteration for computing the Schur decomposition of two matrices [18] starts with setting $Q^{(0)} = I$, $Z^{(0)} = I$. A different suitable initialization follows from a Schur decomposition of just Y_1 and Y_2. At the n-th iteration step, we compute a unitary matrix $Q^{(n)}$ to jointly

make all $Q^{(n)*}(Y_k Z^{(n-1)})$ more upper triangular, and then we compute a unitary matrix $Z^{(n)}$ to make all $(Q^{(n)}Y_k)Z^{(n)}$ more upper triangular. Since it might be hard to find $Q^{(n)}$ or $Z^{(n)}$ to maximize the upper triangularity in each step, it is customary to find only an approximate solution and rely on the outer iteration to provide convergence.

We now describe ways to compute $Q^{(n)}$ (note that $Z^{(n)}$ is obtained in a dual way). At this stage, we have matrices

$$M_k = Q^{(n-1)*} Y_k Z^{(n-1)}, \qquad k = 1, \cdots d$$

not yet upper triangular, and and we have to find an update unitary matrix Q that lowers the below-diagonal norm of $Q^* M_1, \cdots, Q^* M_d$.

Jacobi-type techniques consist of two ingredients: a visiting scheme ("sweep"), in which all below-diagonal entries (i, j), $i > j$ are selected in some order, and an update scheme, where we apply a 2×2 Givens rotation acting on rows i and j. The update scheme can have two policies. If we follow the classical QZ-type Jacobi iteration [14], we minimize the sum of the squared norms of the entries (i, j) after the rotations. The corresponding rotation is computed from

$$U^* \left[\begin{array}{c|c|c|c} M_1(j,j) & M_2(j,j) & \cdots & M_d(j,j) \\ M_1(i,j) & M_2(i,j) & \cdots & M_d(i,j) \end{array}\right] = \left[\begin{array}{c|c|c|c} * & * & \cdots & * \\ \varepsilon_1 & \varepsilon_2 & \cdots & \varepsilon_d \end{array}\right] \tag{5.10.4}$$

If we look at an SVD of the $2\times d$ matrix at the left hand side, it is seen that the smallest $\sum \varepsilon_k^2$ we can obtain is equal to the smallest singular value squared, and the corresponding rotation is the U-factor of the SVD. Alternatively, we can compute U from an eigenvalue decomposition of the 2×2 matrix obtained from squaring the above. Hence, there is a closed-form description of the optimal 2×2 rotation.

A sweep consists in selecting the pivots (i, j) in column-wise ordering, e.g.,

$$M_k = \begin{bmatrix} \times & \times & \times & \times & \times \\ \cdot_1 & \times & \times & \times & \times \\ \cdot_2 & \cdot_5 & \times & \times & \times \\ \cdot_3 & \cdot_6 & \cdot_8 & \times & \times \\ \cdot_4 & \cdot_7 & \cdot_9 & \cdot_{10} & \times \end{bmatrix}$$

Since the rotations are unitary, it is seen that when we are performing rotations in column j, the below-diagonal norm of the previous columns do not change. However, the below-diagonal norm of future columns may increase, but this will be reduced in later rotations.

The above rotation scheme cannot guarantee that the below-diagonal norm will be smaller at the end of a sweep (though this is very likely). Many authors therefore propose a scheme in which the effect of a rotation in the (i, j)-plane on the below-diagonal norm is computed, such that this norm is minimized. This is obtained by looking at (for $i > j$)

$$U^* \left[\begin{array}{c|c|c|c} M_1(j,j)\cdots M_1(j,i-1) & \cdots & M_d(j,j)\cdots M_d(j,i-1) \\ M_1(i,j)\cdots M_1(i,i-1) & \cdots & M_d(i,j)\cdots M_d(i,i-1) \end{array}\right] = \left[\begin{array}{c|c|c|c} *\cdots * & \cdots & *\cdots * \\ \varepsilon\cdots\varepsilon & \cdots & \varepsilon\cdots\varepsilon \end{array}\right]$$

The solution is again given (even in closed form) by an SVD in the same way as before. In exchange for its larger complexity, this scheme has the property that the below-diagonal

Given d matrices $\{Y_k\}$, each of size $d \times d$, find T such that $T^{-1}Y_kT^{-*}$ is approximately diagonal.

1. *Init*: $Q^{(0)}, Z^{(0)}$ from a Gen. Schur Decomposition of $(Y_1,\ Y_2)$.
   Set $R_k = Q^{(0)*}Y_kZ^{(0)}$ $(k = 1, \cdots, d)$
2. Until convergence:
   - *Q-sweep:* for $j = 1, \cdots, d-1$; for $i = j, \cdots, d$
     - SVD: $U\Sigma V^* := \left[\begin{array}{c|c|c|c} R_1(j,j) & R_2(j,j) & \cdots & R_d(j,j) \\ R_1(i,j) & R_2(i,j) & \cdots & R_d(i,j) \end{array}\right]$
     - Apply U^* to rows j and i of $R_1, \cdots, R_d$
   - *Z-sweep*: for $j = d, \cdots, 2$; for $i = j-1, \cdots, 1$
     - SVD: $U\Sigma V^* := \left[\begin{array}{cc} R_1(j,j) & R_1(i,j) \\ \hline \vdots & \vdots \\ \hline R_d(j,j) & R_d(i,j) \end{array}\right]$
     - Apply V to columns j and i of $R_1, \cdots, R_d$
3. $R := [\mathbf{r}_1 \quad \mathbf{r}_2 \ \cdots \ \mathbf{r}_d]$ where $\mathbf{r}_k = \mathrm{diag}[R_k]$
   $A := R^{-1}$, with entries α_{ij}.
4. for $j = 1, \cdots, d$:
   SVD on $\sum_j \alpha_{ij}Y_j$, set $\mathbf{t}_j$ equal to dominant singular vector
   $T = [\mathbf{t}_1, \cdots, \mathbf{t}_d]$

Figure 5.9. Joint diagonalization algorithm

norm cannot increase. Hence, it will converge monotonically to a local minimum. However, in practice the performance of both schemes is quite similar. Convergence is fast, usually quadratically within 3–5 iterations if initialized by a Schur decomposition of two matrices. It should be noted that there are many other possible schemes, e.g., by intertwining the 2×2 Q and Z-rotations or computing them jointly. An experimental performance study made by DeLathauwer [29] shows that the performance of all schemes is grosso modo the same, with the monotonic scheme being somewhat more susceptible to local minima.

The overall joint diagonalization algorithm is summarized in figure 5.9.

Other joint diagonalization problems and algorithms

Many blind source separation problems in signal processing, and elsewhere, lead to joint diagonalization problems. In the literature, we find problems of the form

$$\begin{array}{llcll} (a) & E_i & = & T\Lambda_i T^{-1} & \rightarrow \quad QR_iQ^* \\ (b) & E_i & = & T\Lambda_i T^{*} & \rightarrow \quad QR_iZ^* \\ (c) & E_i & = & A\Lambda_i T^{*} & \rightarrow \quad QR_iZ^* \end{array} \tag{5.10.5}$$

where A and T are square invertible and all Λ_i are diagonal. Problem (c) occurs in multidimensional ESPRIT-type algorithms [49], [54], [22], and also in multilinear source separation problems (called PARAFAC, see chapter 6 of this volume) where we have a data model of the form $\mathbf{x} = \sum_1^d \mathbf{a}_i \otimes \mathbf{l}_i \otimes \mathbf{t}_i$ [41]. Problem (b) is called a Simultaneous Diagonalization by Congruence problem and results in ACMA and JADE [11], and several other cumulant-based algorithms for separating non-Gaussian signals as well as second order techniques for separating sources based on differences in their spectra [5]. Problem (a) is called a Simultaneous Diagonalization by Similarity problem, and is in signal processing applications usually derived from problem (c) by premultiplying with E_0^{-1}, thus eliminating A.

One solution strategy for problem (c) is to solve it iteratively, using alternating least squares [7], [41]. In most cases, however, QR-factorizations for T, T^* and A are introduced, so that we obtain the modified problems at the right of (5.10.5), in which Q and Z are unitary, and the R_i are upper triangular. Another possibility is to assume that T in problem (a) or (b) is unitary, which happens asymptotically if we perform prewhitening. Thus, we arrive at the following set of Simultaneous Schur Decompositions:

$$\begin{array}{lll} (d) & E_i & = & Q\Lambda_i Q^* \\ (e) & E_i & = & QR_i Q^* \\ (f) & E_i & = & QR_i Z^* \end{array} \tag{5.10.6}$$

Overviews of several such problems are given in [8], [12], [15]. For this set of problems, Jacobi-type iterations can be introduced. Problem (d) has received most attention [8], [11], [5], and 2×2 rotations to minimize off-diagonal norms can be obtained in closed form [10]. Alternatively, the problem can be solved using isospectral flows [12]. Problem (f) has been studied in the previous subsection, and elsewhere [50], [30], [29], [1], and also admits closed-form expressions for the 2×2 rotations. Finally, problem (e) has been studied, but no exact closed-form solutions have been reported [23], [1]. However, since this problem is usually derived from problem (a), which in turn is usually obtained from problem (c), it is often possible to consider problem (f) instead.

5.11 Concluding remarks

In this chapter we have described the ACMA: a deterministic block-algorithm for solving the constant modulus problem. Although more complex, it has major advantages over iterative CMAs, e.g.,

– a modest requirement on the number of samples,

– all beamforming vectors for all sources are found in one shot, as the eigenvectors of a generalized eigenvalue problem,

– the solutions are found algebraically and reliably, hence no convergence problems as in iterative CMAs.

We have shown that ACMA converges to the Wiener solution (in samples or SNR), whereas the minima of the CMA(2,2) cost function only have this property if there is no noise or the mixing matrix is orthogonal. Furthermore, we have derived a modification, WACMA, which is close to the zero-forcing solution if the noise power is small (SNR better than 10 dB). We have made a performance comparison to the related JADE algorithm, which separates independent non-Gaussian sources based on their non-zero kurtosis. The conclusion is not unequivocally, because JADE converges to a zero-forcing beamformer asymptotically in the number of samples, but not in SNR. Applied to constant-modulus sources, ACMA has almost always better SINR-performance and WACMA has almost always better SIR-performance.

There are several extensions of this type of algorithm. We described an extension to binary sources. Similar algorithms have been derived for sources that are either zero or constant modulus [53], and the separation of binary sources with unknown residual carriers [48]. The algorithm can be used as the second step in several blind FIR-MIMO channel identification/separation algorithms [32], [52]. Another application is in direction finding of multiple sources: WACMA can be used to estimate individual array response vectors $\mathbf{a}_i$, from which (in the absence of multipath) the directions can be estimated. The algorithm and performance bounds have been published in [31]. A recent overview that places many of these algebraic techniques for blind source separation into context can be found in [47].

BIBLIOGRAPHY

[1] K. Abed-Meraim and Y. Hua, "A least-squares approach to joint Schur decomposition," in *Proc. IEEE ICASSP*, pp. 2541–2544, 1998.

[2] B.G. Agee, "The least-squares CMA: A new technique for rapid correction of constant modulus signals," in *Proc. IEEE ICASSP*, (Tokyo), pp. 953–956, 1986.

[3] K. Anand, G. Mathew, and V.U. Reddy, "Blind separation of multiple co-channel BPSK signals arriving at an antenna array," *IEEE Signal Proc. Letters*, vol. 2, pp. 176–178, Sept. 1995.

[4] K. Anand and V.U. Reddy, "Maximum likelihood estimation of constellation vectors for blind separation of co-channel BPSK signals and its performance analysis," *IEEE Trans. Signal Proc.*, vol. 45, pp. 1736–1741, July 1997.

[5] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, "A blind source separation technique using second-order statistics," *IEEE Trans. Signal Proc.*, vol. 45, pp. 434–444, Feb. 1997.

[6] A. Belouchrani and J.-F. Cardoso, "Maximum likelihood source separation for discrete sources," in *Proc. Eusipco*, pp. 768–771, 1994.

[7] R. Bro, N. Sidiropoulos, and G. Giannakis, "A fast least squares algorithm for separating trilinear mixtures," in *Proc. Workshop on Independent Component Analysis and Blind Source Separation (ICA99)*, (Aussois (Fr)), pp. 289–294, Jan. 1999.

[8] A. Bunse-Gerstner, R. Byers, and V. Mehrmann, "Numerical methods for simultaneous diagonalization," *SIAM J. Matrix Anal. Appl.*, vol. 4, pp. 927–949, 1993.

[9] J.-F. Cardoso, S. Bose, and B. Friedlander, "On optimal source separation based on second and fourt order cumulants," in *Proc. 8th IEEE Signal Proc. Workshop on Stat. Signal Array Proc.*, (Corfu (Gr)), pp. 198–201, 1996.

[10] J.-F. Cardoso and A. Souloumiac, "Jacobi angles for simultaneous diagonalization," *SIAM J. Matrix Anal. Appl.*, vol. 17, no. 1, pp. 161–164, 1996.

[11] J.F. Cardoso and A. Souloumiac, "Blind beamforming for non-Gaussian signals," *IEE Proc. F (Radar and Signal Processing)*, vol. 140, pp. 362–370, Dec. 1993.

[12] M.T. Chu, "A continuous Jacobi-like approach to the simultaneous reduction of real matrices," *Lin. Alg. Appl.*, vol. 147, pp. 75–96, 1991.

[13] P. Comon, "Independent component analysis, a new concept?," *Signal Processing*, vol. 36, pp. 287–314, Apr. 1994.

[14] P.J. Eberlein, "On the Schur decomposition of a matrix for parallel computation," *IEEE Trans. Comp.*, vol. C-36, pp. 167–174, 1987.

[15] B.D. Flury and B.E. Neuenschwander, "Simultaneous diagonalization algorithms with applications in multivariate statistics," in *Approximation and Computation* (R.V.M. Zahar, ed.), pp. 179–205, Basel: Birkhäuser, 1995.

[16] R.W. Gerchberg and W.O. Saxton, "A practical algorithm for the determination of phase from image and diffraction plane pictures," *Optik*, vol. 35, pp. 237–246, 1972.

[17] D.N. Godard, "Self-recovering equalization and carrier tracking in two-dimensional data communication systems," *IEEE Trans. Communications*, vol. 28, pp. 1867–1875, Nov. 1980.

[18] G. Golub and C.F. Van Loan, *Matrix Computations*. The Johns Hopkins University Press, 1989.

[19] R. Gooch and J. Lundell, "The CM array: An adaptive beamformer for constant modulus signals," in *Proc. IEEE ICASSP*, (Tokyo), pp. 2523–2526, 1986.

[20] R.P. Gooch and B.J. Sublett, "Joint spatial and temporal equalization in a decision-directed adaptive antenna system," in *22-nd Asilomar Conf. on Signals, Systems and Computers*, pp. 255–259 vol.1, Maple Press, 1988.

[21] Ming Gu and Lang Tong, "Geometrical characterizations of constant modulus receivers," *IEEE Trans. Signal Processing*, vol. 47, pp. 2745–2756, Oct. 1999.

[22] M. Haardt, *Efficient One-, Two-, and Multidimensional High-Resolution Array Signal Processing*. PhD thesis, TU München, Munich, Germany, 1997.

[23] M. Haardt and J.A. Nossek, "Simultaneous Schur decomposition of several nonsymmetric matrices to achieve automatic pairing in multidimensional harmonic retrieveal problems," *IEEE Trans. Signal Proc.*, vol. 46, pp. 161–169, Jan. 1998.

[24] B. Halder, B.C. Ng, A. Paulraj, and T. Kailath, "Unconditional maximum likelihood approach for blind estimation of digital signals," in *Proc. IEEE ICASSP*, vol. 2, (Atlanta (GA)), pp. 1081–1084, May 1996.

[25] L.K. Hansen and G. Xu, "A hyperplance-based algorithm for the digital co-channel communications problem," *IEEE Trans. Informat. Th.*, vol. 43, pp. 1536–1548, Sept. 1997.

[26] R. Johnson, P. Schniter, T.J. Endres, J.D. Behm, D.R. Brown, and R.A. Casas, "Blind equalization using the constant modulus criterion: a review," *Proc. IEEE*, vol. 86, pp. 1927–1950, Oct. 1998.

[27] A.V. Keerthi, A. Mathur, and J.J. Shynk, "Misadjustment and tracking analysis of the constant modulus array," *IEEE Tr. Signal Processing*, vol. 46, pp. 51–58, Jan. 1998.

[28] M.G. Larimore and J.R. Treichler, "Convergence behavior of the constant modulus algorithm," in *Proc. IEEE ICASSP*, pp. 13–16 vol.1, 1983.

[29] L. De Lathauwer, *Signal Processing Based on Multilinear Algebra*. PhD thesis, KU Leuven, Leuven, Belgium, 1997.

[30] L. De Lathauwer, B. De Moor, and J. Vandewalle, "Independent component analysis based on higher-order statistics only," in *Proc. IEEE SP Workshop on Stat. Signal Array Proc.*, (Corfu, Greece), pp. 356–359, 1996.

[31] A. Leshem and A.J. van der Veen, "Direction of arrival estimation for constant modulus signals," *IEEE Tr. Signal Proc.*, vol. 47, Nov. 1999.

[32] H. Liu and G. Xu, "Closed-form blind symbol estimation in digital communications," *IEEE Trans. Signal Proc.*, vol. 43, pp. 2714–2723, Nov. 1995.

[33] A. Mathur, A.V. Keerthi, J.J. Shynk, and R.P. Gooch, "Convergence properties of the multistage constant modulus array for correlated sources," *IEEE Tr. Signal Processing*, vol. 45, pp. 280–286, Jan. 1997.

[34] C. Moler and G. W. Stewart, "An algorithm for generalized matrix eigenvalue problems," *SIAM Journal on Numerical Analysis*, vol. 10, pp. 241–256, 1973.

[35] T. Nguyen and Z. Ding, "Blind CMA beamforming for narrowband signals with multipath arrivals," *Int. J. Adaptive Control and Signal Processing*, vol. 12, pp. 157–172, Mar. 1998.

[36] 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.

[37] P. Pelin, *Space-time Algorithms for Mobile Communications*. PhD thesis, Chalmers University, Goteborg (S), Aug. 1989.

[38] Y. Sato, "A method of self-recovering equalization for multilevel amplitude-modulation systems," *IEEE Trans. Communications*, vol. 23, pp. 679–682, June 1975.

[39] J. Sheinvald, "On blind beamforming for multiple non-Gaussian signals and the Constant-Modulus Algorithm," *IEEE Trans. Signal Proc.*, vol. 46, pp. 1878–1885, July 1998.

[40] 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.

[41] N. Sidiropoulos, R. Bro, and G. Giannakis, "Parallel factor analysis in sensor array processing," *subm. IEEE Trans. Signal Proc.*, 1999.

[42] S. Talwar, M. Viberg, and A. Paulraj, "Blind estimation of multiple co-channel digital signals arriving at an antenna array," in *27-th Asilomar Conf. Signals, Syst. Comp.*, pp. 349–353 vol.1, IEEE, 1993.

[43] S. Talwar, M. Viberg, and A. Paulraj, "Blind estimation of synchronous co-channel digital signals using an antenna array. Part I: Algorithms," *IEEE Trans. Signal Proc.*, vol. 44, pp. 1184–1197, May 1996.

[44] 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.

[45] 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.

[46] A.J. van der Veen, "Analytical method for blind binary signal separation," *IEEE Trans. Signal Proc.*, vol. 45, pp. 1078–1082, Apr. 1997.

[47] A.J. van der Veen, "Algebraic methods for deterministic blind beamforming," *Proc. IEEE*, vol. 86, pp. 1987–2008, Oct. 1998.

[48] A.J. van der Veen, "Blind separation of BPSK sources with residual carriers," *Signal Processing*, vol. 73, pp. 67–79, Jan. 1999.

[49] A.J. van der Veen, P.B. Ober, and E.F. Deprettere, "Azimuth and elevation computation in high resolution DOA estimation," *IEEE Trans. Signal Processing*, vol. 40, pp. 1828–1832, July 1992.

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

[51] A.J. van der Veen and A. Paulraj, "Singular value analysis of space-time equalization in the GSM mobile system," in *Proc. IEEE ICASSP*, (Atlanta (GA)), pp. 1073–1076, May 1996.

[52] A.J. van der Veen, S. Talwar, and A. Paulraj, "A subspace approach to blind space-time signal processing for wireless communication systems," *IEEE Trans. Signal Proc.*, vol. 45, pp. 173–190, Jan. 1997.

[53] A.J. van der Veen and J. Tol, "Separation of zero/constant modulus signals," in *Proc. IEEE ICASSP*, (Munich (Germany)), pp. 3445–3448, Apr. 1997.

[54] A.J. van der Veen, M.C. Vanderveen, and A. Paulraj, "Joint angle and delay estimation using shift-invariance techniques," *IEEE Tr. Signal Processing*, vol. 46, pp. 405–418, Feb. 1998.

[55] M. Wax and Y. Anu, "A Least-Squares approach to blind beamforming," *IEEE Trans. Signal Proc.*, vol. 47, pp. 231–234, Jan. 1999.

[56] M. Wax and J. Sheinvald, "A least-squares approach to joint diagonalization," *IEEE Signal Proc. Letters*, vol. 4, pp. 52–53, Feb. 1997.

[57] H.H. Zeng, Lang Tong, and C.R. Johnson, "Relationships between the constant modulus and Wiener receivers," *IEEE Trans. Inform. Theory*, vol. 44, pp. 1523–1538, July 1998.