in Handbook on Antennas in Wireless Communications, edt. Lal Godara, CRC Press, Boca Raton, FL, 2001

Blind Channel Identification and Source Separation in SDMA Systems

Victor Barroso*, João Xavier*, and José M.F. Moura†

*Instituto Superior Técnico, Instituto de Sistemas e Robótica

Av. Rovisco Pais, 1049–001 Lisboa, Portugal

†Dept. of Electrical and Computer Engineering

Carnegie Mellon University, 5000 Forbes Avenue Pittsburgh PA 15213-3890, USA

Contents

LIST OF ACRONYMS

ACMA	Analytical Constant Modulus Algorithm
ATM	Admissible Transformation Matrix
BSS	Blind Source Separation
CDMA	Code Division Multiple Access
CFC_2	Closed–Form Correlative Coding
CM	Constant Modulus
DOA	Direction Of Arrival
EM	Expectation–Maximization
EVD	Eigen Value Decomposition
FDMA	Frequency Division Multiple Access
FM	Frequency Modulation
FSK	Frequency Shift Keying
GDA	Geodesic Descent Algorithm
GML	Generalized Maximum Likelihood
GSA	Gerchberg–Saxton Algorithm
HOS	Higher Order Statistics
ILSE	Iterative Least Squares with Enumeration
ILSP	Iterative Least Squares with Projection
IML	Iterative Maximum Likelihood
ISI	Inter–Symbol Interference
LP	Linear Programming
LS	Least Squares
MAP	Maximum A Posteriori
MIMO	Multiple–Input Multiple–Output
ML	Maximum Likelihood
OPDA	Outer–Product Decomposition Algorithm
PAM	Pulse Amplitude Modulation
PM	Phase Modulation
PSK	Phase Shift Keying
QAM	Quadrature Amplitude Modulation
QP	Quadratic Programming
SDMA	Space Division Multiple Access
SIMO	Single–Input Multiple–Output
SML	Stochastic Maximum Likelihood
SNR	Signal to Noise Ratio
SOS	Second Order Statistics
SVD	Singular Value Decomposition
TDMA	Time Division Multiple Access
TICC	Transmitter Induced Conjugate Cyclostationarity

1 INTRODUCTION

In this chapter we address the problem of discriminating radio sources in the context of cellular mobile wireless digital communications systems. The main purpose of solving this problem is to increase the overall capacity of these systems. Usually, the sources are discriminated in frequency, time, or code. In the case of frequency division multiple access (FDMA) systems, each user has assigned a different frequency band, while in time division multiple access (TDMA) systems the users can share the same frequency band, but transmit during disjoint time slots. Finally, code division multiple access (CDMA) systems are based in spread spectrum techniques, where a different spreading code is assigned to each user. The spreading codes are chosen to be approximately orthogonal so that the sources present share simultaneously the same frequency bandwidth.

Here, we will consider space division multiple access (SDMA) systems. These systems utilize the geographic diversity of the users location in a given cell at a given time interval. Suppose that the antenna of the cell base station has a fixed multi–beam beampattern. Then, for practical purposes, users illuminated by beam i will not interfere with those illuminated by beam j, even if the users in both beams transmit at the same time, and share simultaneously the same frequency band and/or the same set of orthogonal codes. In this case, a cell with a four beam antenna will have its capacity increased by a factor that is ideally four. This basic idea of sectoring a cell in several spatially disjoint smaller cells based on a single fixed multi–beam base station antenna can be translated with considerable gains into a more evolved and flexible system architecture. This relies on the concept of smart antennas and gives rise to what can be denoted smart SDMA systems, see the pictorial illustration in figure 1. A smart antenna

Figure 1: The concept of smart SDMA systems

is an antenna array whose beampattern is controled electronically and/or numerically so as to illuminate the desired sources and cancel out the interferences. Usually, this is done by pointing the beam into the direction of arrival (DOA) of the desired wavefront(s), while forcing deep nulls of the beampattern at

the interferences' DOAs. Classically, this concept of smart antenna involves well known DOA estimation and adaptive/blind beamforming algorithms, see [43, 31, 20] and [11, 12, 27, 21]. Here, we consider a different approach, which will be clarified in subsection 1.1, where we set-up the model of the antenna array observations data. Before doing that, we introduce notation adopted in the chapter.

NOTATION

$\mathbb{N}$, $\mathbb{Z}$, $\mathbb{R}$, and $\mathbb{C}$ denote the set of natural, integer, real, and complex numbers, respectively. Matrices (uppercase) and (column/row) vectors are in boldface type. $\mathbb{C}^{n\times m}$ and $\mathbb{C}^n$ denote the set of $n \times m$ matrices and the set of n-dimensional column vectors with complex entries, respectively. The notations $(\cdot)^*$, $(\cdot)^T$, $(\cdot)^H$, $(\cdot)^\dagger$, and $\mathrm{tr}\,(\cdot)$ stand for the conjugate, transpose, the Hermitean, the Moore-Penrose pseudo-inverse, and the trace operator, respectively; $||\boldsymbol{A}|| = \sqrt{\mathrm{tr}\left(\boldsymbol{A}^H\boldsymbol{A}\right)}$ denotes the Frobenius norm. The symbols $\boldsymbol{I}_n$, $\boldsymbol{0}_{n\times m}$, and $\boldsymbol{J}_n$ stand for the $n \times n$ identity, the $n \times m$ all-zero, and the $n \times n$ forward-shift (ones in the first lower diagonal) matrices, respectively. When the dimensions are clear from the context, the subscripts are dropped. The direct sum or diagonal concatenation of matrices is represented by $\mathrm{diag}\,(\boldsymbol{A}_1, \boldsymbol{A}_2, \cdots, \boldsymbol{A}_m)$; for $\boldsymbol{A} \in \mathbb{C}^{n\times m}$, $\mathrm{vec}\,(\boldsymbol{A}) \in \mathbb{C}^{nm}$ consists of the columns of $\boldsymbol{A}$ stacked from left to right; and $\otimes$ represents the Kronecker product. For $\boldsymbol{A} \in \mathbb{C}^{n\times n}$, $\sigma\,(\boldsymbol{A}) = \{\,\lambda_1,\, \lambda_2,\, \cdots,\, \lambda_n\,\}$ denotes its spectrum, i.e., the set of its eigenvalues (including multiplicities). For $\boldsymbol{A} \in \mathbb{C}^{n\times m}$, we let $\mathrm{row}(\boldsymbol{A})$ $(\mathrm{col}(\boldsymbol{A}))$ denote the subspace of $\mathbb{C}^m$ $(\mathbb{C}^n)$ spanned by the rows (columns) of $\boldsymbol{A}$.

1.1 ARRAY DATA MODEL

Consider P independent digital sources, where each one generates the base band signal

$$r_p(t) = \sum_{k=-\infty}^{+\infty} s_p(k) u_p(t - kT), \tag{1}$$

where $u_p(t)$ is a unit energy shaping pulse, T is the baud period and $\{s_p(k)\}_{k=-\infty}^{+\infty}$ is the information sequence of independent and equally like symbols generated by the source p. These symbols are taken from a generic digital source alphabet $\mathcal{A}$. We assume that the information bearing signal $r_p(t)$ modulates a radio carrier frequency ω_c. The resulting digital modulated signal is received by an antenna array of N_a omnidirectional sensors (antenna elements). At the array sensor n, the received signal has the complex envelope

$$y_{p_n}(t) = \sum_{m=1}^{M_p} \alpha_{mp} e^{-j\omega_c \tau_{mp}} r_p(t - \tau_{mp} - \Delta_{mp}^{(n)}) e^{-j\omega_c \Delta_{mp}^{(n)}}, \tag{2}$$

where M_p is the number of propagation paths, α_{mp} and τ_{mp} are the corresponding attenuations and travel time path delay, respectively, and $\Delta_{mp}^{(n)}$ measures the inter–sensor propagation delay with respect to a given reference sensor. For each specific array and array/source geometry, the inter–sensor delays can be parameterized in terms of the DOAs.

Equation (2) is now rewritten into a more compact framework. We first define the overall channel impulse response between source p and sensor n, including the shaping pulse, as the time convolution

$$h_{pn}(t) = u_p(t) \star h_{\mathrm{cha}_{pn}}(t),$$

where the sensor/channel impulse responses are

$$h_{\mathrm{cha}_{pn}}(t) = \sum_{m=1}^{M_p} \alpha_{mp} e^{-j\omega_c(\tau_{mp}+\Delta_{mp}^{(n)})} \delta(t - \tau_{mp} - \Delta_{mp}^{(n)}).$$

Then equation (2) is rewritten as

$$y_{p_n}(t) = \sum_{k=-\infty}^{+\infty} h_{pn}(t - kT) s_p(k). \tag{3}$$

This signal is time sampled with a sampling period T_s such that $T/T_s = J \geq 1$ is an integer. Then, assuming that the overall channel impulse response spans L_p baud periods, we write

$$y_{p_n}(kT + jT_s) = \sum_{l=0}^{L_p-1} h_{pn}(lT + jT_s) s_p(k - l), \qquad j = 0, 1, \ldots, J-1. \tag{4}$$

Here L_p determines the temporal extension of the inter-symbol interference (ISI) induced by the physical channel used by source p. The jth-sample of the received signal for the kth-baud period is given by equation (4) as a convolution over the variable l, i.e., over the multipath structure for that sample.

To obtain a compact expression for the received signal for an array of N_a sensors, P sources, and J samples per baud period we stack succesively the samples in equation (4). We first consider a single source p. Start by stacking the received samples in equation (4) over the variable j to form the J-dimensional vector

$$\boldsymbol{y}_{p_n}(k) = \left[y_{p_n}(kT) \cdots y_{p_n}(kT + (J-1)T_s)\right]^T, \qquad n = 1, \cdots, N_a.$$

Then stack these N_a vectors of dimension J into the N-dimensional vector, where $N = J \times N_a$,

$$\boldsymbol{y}_p(k) = \left[\boldsymbol{y}_{p_1}(k)^T, \cdots, \boldsymbol{y}_{p_{N_a}}(k)^T\right]^T, \qquad p = 1, \cdots, P. \tag{5}$$

Likewise, we stack the delayed replicas of the information bearing signal for baud period k into the L_p-dimensional vector

$$\boldsymbol{s}_p(k) = \left[s_p(k)\ s_p(k-1) \cdots s_p(k - L_p + 1)\right]^T,$$

and the channel impulse response for sample j from source p to sensor n into the row L_p-dimensional vector

$$\boldsymbol{h}_{pn}(j) = \left[h_{pn}(jT_s)\ h_{pn}(T + jT_s) \cdots h_{pn}((L_p - 1)T + jT_s)\right].$$

Define the $J \times L_p$-block matrices that collect for the J-samples these channel impulse responses from source p to sensor n

$$\boldsymbol{H}_{pn} = \left[\boldsymbol{h}_{pn}^T(0)\ \boldsymbol{h}_{pn}^T(1) \cdots \boldsymbol{h}_{pn}^T(J-1)\right]^T,$$

and put together these block matrices for all N_a-sensors of the array to define the $N \times L_p$ channel matrix for source p as

$$\boldsymbol{H}_p = \left[\boldsymbol{H}_{p1}^T\ \boldsymbol{H}_{p2}^T \cdots \boldsymbol{H}_{pN}^T\right]^T.$$

We can now find a concise expression for the $N = JN_a$-dimensional vector $\boldsymbol{y}_p(k)$. Using the notation just introduced, the convolution in equation (4) becomes

$$y_{p_n}(kT + jT_s) = \boldsymbol{h}_{pn}(j) \cdot \boldsymbol{s}_p(k).$$

The received signal $\boldsymbol{y}_p(k)$ for source p at all the array sensors in equation (5) is given by

$$\boldsymbol{y}_p(k) = \boldsymbol{H}_p \boldsymbol{s}_p(k). \tag{6}$$

Notice that each block matrix $\boldsymbol{H}_p$ determines the multipath propagation channel, including the array response, used by source p. Therefore all the effects, such as the ISI, induced by that channel are embedded in $\boldsymbol{H}_p$. Naturally, sources with distinct cell locations will use distinct physical channels. This means that each $\boldsymbol{H}_p$ acts like a source signature.

It is now easy to generalize the single user model in equation (6) to the case of multiple users. Letting

$$\boldsymbol{H} = [\boldsymbol{H}_1\ \boldsymbol{H}_2 \cdots \boldsymbol{H}_P] \in \mathbb{C}^{N \times M},$$

with $M = \sum_{p=1}^{P} L_p$, and the M–dimensional information signal vector

$$\boldsymbol{s}(k) = \left[\boldsymbol{s}_1^T(k)\ \boldsymbol{s}_2^T(k) \cdots \boldsymbol{s}_P^T(k)\right]^T,$$

we can define the N–dimensional vector of array observations as

$$\boldsymbol{x}(k) = \boldsymbol{H}\boldsymbol{s}(k) + \boldsymbol{n}(k),\ k = 1, 2, \ldots \tag{7}$$

where the N–dimensional complex vector $\boldsymbol{n}(k)$ represents the array observations noise at time instant (baud period) k. This will be assumed to be a zero mean stationary white Gaussian vector process with covariance matrix $\sigma^2 \boldsymbol{I}_N$.

Eq. (7) represents the array snapshot at the kth-symbol period. If we work with K symbols, and assume that the channel matrix $\boldsymbol{H}$ is constant during the observation time interval of length K symbol periods, we can collect the K array snapshots in the $(N \times K)$ matrix

$$\boldsymbol{X} = \left[\ \boldsymbol{x}(1) \quad \boldsymbol{x}(2) \quad \cdots \quad \boldsymbol{x}(K)\ \right].$$

Then eq. (10) is compactly written as

$$\boldsymbol{X} = \boldsymbol{H}\boldsymbol{S} + \boldsymbol{N}, \tag{8}$$

where $\boldsymbol{S}$ and $\boldsymbol{N}$ are, respectively, the $(P \times K)$ matrix of the binary sequences of the sources and the $(N \times K)$ noise matrix,

$$\boldsymbol{S} = \left[\begin{array}{cccc} \boldsymbol{s}(1) & \boldsymbol{s}(2) & \cdots & \boldsymbol{s}(K) \end{array} \right] \text{ and } \boldsymbol{N} = \left[\begin{array}{cccc} \boldsymbol{n}(1) & \boldsymbol{n}(2) & \cdots & \boldsymbol{n}(K) \end{array} \right]. \tag{9}$$

This is the data model that we will use.

Convolutive mixture model Formally, the first term on the right–hand side of eq. (7) or eq. (8) represents a finite mixture of P time convolutions. This model is usually denoted as a multiple input–multiple output (MIMO) convolutive mixture model.

Instantaneous mixture model In several real scenarios, such as in pico–cells, the propagation delays in (2) are much smaller than the baud period T. This means that the effect of ISI can be ignored. In these situations, $L_p = 1,\ p = 1, \ldots, P$, and the generic convolutive mixture model in (7) or eq. (8) degenerates into an instantaneous mixture model. It is clear, in any case, that we can still formally use the same representation as in (8).

1.2 BLIND SOURCE SEPARATION

We use the same formulation for the source separation problem for the two models described in the last subsection:

Given the array data set $\boldsymbol{X}$, as defined in eq. (9), where the channel mixture matrix $\boldsymbol{H}$ is unknown, find the P source information sequences $s_p(k)$, $k = 1, \ldots, K$, i.e., *find $\boldsymbol{S}$.*

The approach that we take to address this problem is **blind**, since we will not rely on any apriori knowledge about the channel. The solutions presented in this chapter only use the information in one or several of the following: **(i)** the array data model; and/or **(ii)** the noise statistics; and/or **(iii)** features of the source data sequences, such as, their statistics and properties of the alphabet of symbols they use.

With respect to the two mixture models considered in the last sub-section – instantaneous and convolutive mixture – we can identify two distinct sub–problems: blind source separation and blind equalization (ISI cancellation). With convolutive mixtures, these two sub–problems need to be solved. In the simpler context of instantaneous mixtures, the ISI effect is absent and only the source separation sub–problem is involved.

We present below several methods to solve these problems. In all the approaches considered, we assume that the channel matrix $\boldsymbol{H}$ is time invariant along the duration $T_{\text{obs}} = KT$ of K baud periods in the observation interval. For actual wireless channels, we can say that this approximation holds when T_{obs} is not very large when compared with the baud period T. This constrains the number of time samples that can be used to solve any of the sub–problems referred to above and, as a consequence, the type

of method that will be used for each channel. In general, statistical methods are used to identify slow to moderately fast time varying channels, while faster channels are usually identified with deterministic techniques.

1.3 CHAPTER SUMMARY

This chapter describes several methods that can be applied to solve the blind source separation and ISI cancellation problems that we are considering. The presentation includes, by this order, deterministic or zero–order statistics (ZOS), second–order statistics (SOS), and stochastic maximum–likelihood (SML) or infinite–order statistics (IOS) methods. Higher–order statistics (HOS) methods are not considered because of three main reasons: **(i)** the large amount of data necessary to obtain efficient estimates of the HOS is compatible only with very slow varying channels, **(ii)** the computational complexity of the resulting identification algorithms can be very high, and **(iii)** they are very sensitive to the SNR (signal to noise ratio). Except for the stochastic maximum–likelihood (SML) method where we consider only the instantaneous mixture problem, we study both the instantaneous and convolutive mixtures. We present algorithms that have been found highly relevant contributions within each class of methodologies. We include several of our own contributions.

Section 2 is mainly concerned with ZOS methods. We introduce several algorithms that result from a deterministic (ZOS) approach and that are suited when the channels are moderately varying. We describe the Iterative Least–Squares with Projection (ILSP) and the Iterative Least–Squares with Enumeration (ILSE) algorithms. These algorithms exploit the finite alphabet property of digital sources. We also present the Analytical Constant Modulus Algorithm (ACMA). This algorithm exploits a different feature of several digital modulation formats, namely, the equal energy (or constant modulus) of many signaling schemes. Like ILSP and ILSA, the ACMA requires an iterative procedure for which global convergence is not guaranteed, except when the observations are noiseless. In alternative to iterative algorithms and to avoid their convergence problems, it is most important to have available closed–form solutions. We describe a solution that has low computational complexity. This closed–form method is based on a linear coding approach involving the sources data. We finalize the section by introducing the subspace method that solves the ISI cancellation sub–problem.

Section 3 addresses SOS based approaches. Essentially, the SOS are used to derive algorithms based on data prewhitening. This enables techniques that provide analytical closed–form solutions for the two sub–problems that we consider, i.e., for the blind ISI cancellation and source separation problems. Notice that, in general, for white source sequences, the SOS based source separation problem is ill–defined, and HOS techniques are required. The correlative coding approach, introduced in this section, establishes a framework within which it is possible to guarantee a unique solution for the SOS based blind source separation problem.

Section 4 is devoted to SML methods. The algorithms described in section 2 can actually also be

viewed as solutions to a maximum–likelihood (ML) problem where the source signals are deterministic and the noise is white Gaussian. In section 4, we solve the blind source separation problem for stochastic source signals. We restrict attention to instantaneous mixture channels. The solution that we present relies on the Expectation–Maximization (EM) algorithm, where the ML estimation of the channel mixing matrix is used to detect the source symbols based on the maximum a posteriori (MAP) criterion.

The final section of the chapter discusses new solutions to the general source separation problem, based on convex and differential geometry driven optimization techniques. These approaches exploit directly specific geometric features of the data model. The first technique falls in the class of deterministic approaches, and is restricted to high signal-to-noise ratio (SNR) scenarios and binary sources. It requires a small amount of data samples, being especially suited for fast time varying channels. A geometrical convex re-formulation of the blind source separation problem is exploited to resolve the linear mixture of binary users. We also discuss a SOS based semi-blind SML technique for channel identification with white (up to 2nd order) stationary inputs. The likelihood function for the residual unitary matrix is optimized directly over the manifold of orthogonal matrices, by a geodesic descent algorithm. This method can be applied in the cases of analog or digital sources, providing semi–blind and blind solutions, respectively.

2 DETERMINISTIC METHODS

This section discusses deterministic or zero order statistics (ZOS) methods that resolve linear mixtures of binary sources. This class of methods does not make use of the statistics or of the probability structure of the data. However, they take advantage of properties of the digital sources, for example, their finite alphabet property, or they exploit specific characteristics of the signaling/modulation used like the equal energy in signaling schemes. Although the methods described in this section can be extended to more general source alphabets, we will restrict the discussion to antipodal binary alphabets $\mathcal{A} = \{-1, +1\}$. The section addresses both cases of instantaneous and convolutive mixtures.

In subsection 2.1, we present first an identifiability result and then tackle instantaneous mixtures, describing four algorithms: the Iterative Least-Squares with Projection (ILSP) algorithm; the Least-Squares with Enumeration (ILSE) algorithm; the Analytical Constant Modulus Algorithm (ACMA) ; and a closed form solution based on pre-coding. In sub-section 2.2, we study convolutive mixtures and consider a subspace method for ISI cancellation.

2.1 INSTANTANEOUS MIXTURES

In the case of instantaneous mixtures, the transmission is ISI free, and so we take $L_p = 1$, $p = 1, \cdots, P$. The data model is

$$\boldsymbol{x}(k) = \boldsymbol{H}\boldsymbol{s}(k) + \boldsymbol{n}(k), \quad k = 1, 2, \ldots \tag{10}$$

As described in section 1, $\boldsymbol{x}(k)$ and $\boldsymbol{n}(k)$ are N– dimensional vectors denoting, respectively, the array snapshot and the observation noise at time instant k; $\boldsymbol{s}(k)$ is the P-vector of the binary symbols generated by the P sources at time instant k, and $\boldsymbol{H}$ is the $(N \times P)$ channel matrix, i.e., the mixture matrix. The number P of sources is assumed known, or some estimate of it is available. Working with K symbols and grouping the K array snapshots into a single matrix, the data model is as in eq. (8) herein repeated

$$\boldsymbol{X} = \boldsymbol{H}\boldsymbol{S} + \boldsymbol{N}. \tag{11}$$

The matrices in eq. (11) have the following dimensions: $\boldsymbol{X}$ is $N \times K$; $\boldsymbol{H}$ is $N \times P$; $\boldsymbol{S}$ is $P \times K$; and $\boldsymbol{N}$ is $N \times K$. The noise $\boldsymbol{N}$ is zero mean, stationary, white Gaussian.

Problem Formulation. The problem we address is the following: given the noisy data matrix $\boldsymbol{X}$ in (11), find both $\boldsymbol{H}$ and $\boldsymbol{S}$.

Solving this problem needs solving jointly an estimation problem on the continuous "variable" $\boldsymbol{H}$ and a detection problem on the discrete "variable" $\boldsymbol{S}$. Before addressing these problems, we consider the identifiability or the uniqueness of the solution in the absence of noise.

Identifiability. Consider the noiseless, $\boldsymbol{N} = \boldsymbol{0}$, version of (11)

$$\boldsymbol{X} = \boldsymbol{H}\boldsymbol{S}, \tag{12}$$

and assume that $\boldsymbol{H}$ is an arbitrary full column rank matrix. The elements of $\boldsymbol{S}$ belong to the alphabet $\mathcal{A} = \{-1, 1\}$. Under these conditions, any factorization

$$\boldsymbol{X} = \widehat{\boldsymbol{H}}\widehat{\boldsymbol{S}},$$

where

$$\widehat{\boldsymbol{H}} = \boldsymbol{H}\boldsymbol{T} \qquad \text{and} \qquad \widehat{\boldsymbol{S}} = \boldsymbol{T}^{-1}\boldsymbol{S},$$

verifies (12), provided that $\boldsymbol{T}$ is one of the following $(P \times P)$ matrices: non-singular diagonal matrix with ± 1 entries; a permutation matrix; or a product of the two. The matrix $\boldsymbol{T}$ causes two types of ambiguity: **(i)** ordering ambiguity; and **(ii)** sign ambiguity in estimates of the signals. In any case, these ambiguities are easily removed if appropriate coding schemes are used.

In [34], Talwar et al. prove an identifiability theorem that provides a sufficient condition for the existence of an admissible transform matrix (ATM) such as $\boldsymbol{T}$ above. Here, we present their theorem without proof.

Theorem 2.1[Identifiability] *Let $\boldsymbol{X} = \boldsymbol{H}\boldsymbol{S}$ where $\boldsymbol{H}$ is an arbitrary $(N \times P)$ full-rank matrix with $P \leq N$, and $\boldsymbol{S}$ is a $(P \times K)$ full-rank matrix with ± 1 elements. If the columns of $\boldsymbol{S}$ include all the 2^{P-1}*

possible distinct (up to a sign) P-vectors with ±1 elements, then $\boldsymbol{H}$ *and* $\boldsymbol{S}$ *can be uniquely identified up to a* $(P \times P)$ *matrix* $\boldsymbol{T}$ *with exactly one non-zero element* $\{+1, -1\}$ *in each row and column.* □

The probability p that, in K independent array snapshots, the columns of $\boldsymbol{S}$ include all the 2^{P-1} distinct (up to a sign) P-vectors with ±1 elements is also studied in [34]. This probability p is bounded below

$$1 - 2^{P-1}\left(\frac{2^{P-1}-1}{2^{P-1}}\right)^K \leq p \leq 1.$$

Notice that p converges to 1 when K increases indefinitely. It is also clear that, for large values of P, a quite large value of K can be required to guarantee identifiability. This is only apparent. In fact, the *Identifiability Theorem* 2.1 establishes a sufficient condition only. Most likely, smaller values of K will suffice in practical situations.

At this point, we know what is the sufficient condition under which the noiseless factorization problem in eq. (12) can be solved uniquely, up to an admissible transform matrix. In the following paragraphs, we will discuss relevant algorithms that provide that solution.

2.1.1 ILSP and ILSE algorithms

Here, we return to the model described by eq. (11), and assume that the noise is white in both the time and space dimensions, Gaussian, zero mean, and with correlation matrix $E\{\boldsymbol{n}(k)\boldsymbol{n}^H(l)\} = \sigma^2 \boldsymbol{I}_N \delta_{kl}$.

Under these conditions, the Maximum Likelihood (ML) approach is equivalent to the following separable least-squares problem:

$$\left(\widehat{\boldsymbol{H}}, \widehat{\boldsymbol{S}}\right) = \arg\min_{\boldsymbol{H},\boldsymbol{S}} \|\boldsymbol{X} - \boldsymbol{H}\boldsymbol{S}\|_F^2, \tag{13}$$

where the elements of $\boldsymbol{S}$ are assumed deterministic and constrained to take values in $\mathcal{A}$. Recall that $\|\cdot\|_F$ is the Frobenius norm.

This minimization can be done in two steps. Noticing that $\boldsymbol{H}$ is an arbitrary matrix, the minimization in (13) with respect to $\boldsymbol{H}$ is unconstrained, and therefore

$$\widehat{\boldsymbol{H}} = \boldsymbol{X}\boldsymbol{S}^\dagger. \tag{14}$$

Defining the orthogonal projection matrix

$$\boldsymbol{P}_{\boldsymbol{S}}^{\perp} = \boldsymbol{I}_K - \boldsymbol{S}^\dagger \boldsymbol{S}, \tag{15}$$

the substitution of (14) in (13) yields the constrained minimization problem:

$$\widehat{\boldsymbol{S}} = \arg\min_{\boldsymbol{S}\in\mathcal{A}^{P\times K}} \left\|\boldsymbol{X}\boldsymbol{P}_{\boldsymbol{S}}^{\perp}\right\|_F^2, \tag{16}$$

where $\mathcal{A}^{P\times K}$ denotes the set of all the $(P \times K)$ matrices with entries defined in $\mathcal{A}$. The solution to this minimization problem is achieved by enumerating over all possible choices of the binary matrix $\boldsymbol{S}$. It is

clear that the numerical complexity of this enumeration procedure, being exponential with K and P, is prohibitive even for modest values of K and P. In the following paragraphs, we present two iterative block algorithms with lower computational complexity, the Iterative Least-Squares with Projection (ILSP) and the Iterative Least-Squares with Enumeration (ILSE) algorithms, [33, 34].

ILSP Algorithm. For simplicity purposes, assume that the minimization problem in eq. (13) is unconstrained with respect to both matrices $\boldsymbol{H}$ and $\boldsymbol{S}$. Then, starting with an initial estimate $\widehat{\boldsymbol{H}}_0$ of $\boldsymbol{H}$, the minimization of $\left\|\boldsymbol{X}-\widehat{\boldsymbol{H}}_0\boldsymbol{S}\right\|_F^2$ with respect to a continuous $\boldsymbol{S}$ is a simple least-squares problem. Each element of the solution $\overline{\boldsymbol{S}}_1$ is then projected back to the closest discrete value in $\mathcal{A}$, producing the estimate $\widehat{\boldsymbol{S}}_1$. The iterative algorithm runs as follows.

ILSP Algorithm

1. Given $\widehat{\boldsymbol{H}}_0$ for $k=0$
2. $k \leftarrow k+1$
   (a) $\overline{\boldsymbol{S}}_k = \widehat{\boldsymbol{H}}_{k-1}^{\dagger}\boldsymbol{X}$
   (b) $\widehat{\boldsymbol{S}}_k = \text{proj}\left[\overline{\boldsymbol{S}}_k\right]$ $\quad$ $\widehat{\boldsymbol{S}}_k$ is the matrix in $\mathcal{A}^{P\times K}$ closest to $\overline{\boldsymbol{S}}_k$
   (c) $\widehat{\boldsymbol{H}}_k = \boldsymbol{X}\widehat{\boldsymbol{S}}_k^{\dagger}$
3. Repeat step 2. until $\left(\widehat{\boldsymbol{H}}_k, \widehat{\boldsymbol{S}}_k\right) = \left(\widehat{\boldsymbol{H}}_{k-1}, \widehat{\boldsymbol{S}}_{k-1}\right)$

In contrast with the optimal solution, which has exponential complexity in both K and P, the ILSP algorithm has polynomial complexity, more specifically, $KNP+2P^2(K-\frac{P}{3})+NP^2$ and $KNP+2P^2(N-\frac{P}{3})+KP^2$ flops per iteration to compute $\widehat{\boldsymbol{H}}$ and $\widehat{\boldsymbol{S}}$, respectively. Clearly, the overall complexity of the algorithms depends on its convergence rate. This complexity can be controled if the algorithm is appropriately initialized, i.e., started with $\widehat{\boldsymbol{H}}_0$ close to the actual $\boldsymbol{H}$. In the case of low noise observations, this will likely provide local convergence of ILSP to the optimal constrained solution in a reasonable number of iterations. However, if the mixing matrix $\boldsymbol{H}$ is ill conditioned, the ILSP algorithm can diverge, essentially due to the noise enhancement produced by the least squares step to compute $\overline{\boldsymbol{S}}$. This important drawback of ILSP is circumvented by the ILSE algorithm.

ILSE Algorithm. The ILSE algorithm uses one property of the Frobenius norm to reduce the complexity of the enumeration procedure necessary to compute the optimal $\widehat{\boldsymbol{S}}$. According to this property,

$$\min_{\boldsymbol{S}\in\mathcal{A}^{P\times K}}\|\boldsymbol{X}-\boldsymbol{H}\boldsymbol{S}\|_F^2 = \min_{\boldsymbol{s}(1)\in\mathcal{A}^P}\|\boldsymbol{x}(1)-\boldsymbol{H}\boldsymbol{s}(1)\|_F^2 + \cdots + \min_{\boldsymbol{s}(K)\in\mathcal{A}^P}\|\boldsymbol{x}(K)-\boldsymbol{H}\boldsymbol{s}(K)\|_F^2 . \tag{17}$$

This means that, instead of searching among all the 2^{KP} binary matrices $\boldsymbol{S}$, we can simply perform K independent enumerations, each one involving 2^P possible binary vectors $\boldsymbol{s}(\cdot)$:

$$\forall\ \ k=1,\ldots,K: \quad \widehat{\boldsymbol{s}}(k) = \arg\min_{\boldsymbol{s}\in\mathcal{A}^P}\|\boldsymbol{x}(k)-\boldsymbol{H}\boldsymbol{s}\|_F^2 . \tag{18}$$

Except for the minimization with respect to $\boldsymbol{S}$, the ILSE algorithm is very similar to ILSP, each iteration being based on an alternating minimization technique as follows.

ILSE Algorithm

1. Given $\widehat{\boldsymbol{H}}_0$ for $k = 0$
2. $k \leftarrow k + 1$
   (a) With $\boldsymbol{H} \leftarrow \widehat{\boldsymbol{H}}_{k-1}$ in (17), minimize for $\widehat{\boldsymbol{S}}_k$ using (18)
   (b) $\widehat{\boldsymbol{H}}_k = \boldsymbol{X}\widehat{\boldsymbol{S}}_k^{\dagger}$
3. Repeat step 2. until $\left(\widehat{\boldsymbol{H}}_k, \widehat{\boldsymbol{S}}_k\right) = \left(\widehat{\boldsymbol{H}}_{k-1}, \widehat{\boldsymbol{S}}_{k-1}\right)$

The ILSE algorithm has complexity $KNP+2P^2(K-\frac{P}{3})+NP^2$ plus $KN\cdot 2^P(P+1)$ flops per iteration to solve for $\widehat{\boldsymbol{H}}$ and $\widehat{\boldsymbol{S}}$, respectively. Comparing this with the complexity of ILSP, we conclude that ILSE generally requires considerably more flops per iteration to solve for $\widehat{\boldsymbol{S}}$. Contrarily to what happens with the ILSP algorithm, the ILSE algorithm has local uniform convergence and exhibits greater robustness against the observation noise, especially when $\boldsymbol{H}$ is ill conditioned. These are the main advantages of ILSE over ILSP.

A detailed performance analysis of the ILSP and the ILSE algorithms is out of the scope of this chapter. The interested reader may found it in [35]. Here we notice that two alternative algorithms based on successive interference cancellation concepts are introduced in [22]. Like ILSE, the interference cancellation algorithms are at least monotonically convergent to a local minimum, and attain a performance similar to that of ILSE at the complexity cost of ILSP.

2.1.2 Analytical constant modulus algorithm

While ILSP and ILSE take advantage of the finite alphabet property of digital signals, the constant modulus (CM) approach exploits the time invariance of the envelope of many communications signals such as FM and PM in the analog domain, and FSK, PSK, and 4-QAM for digital signals. The concept of modulus restoration was first introduced in the context of blind equalization problems, see [28, 10]. It has also been applied to solve the problem of resolving instantaneous linear mixtures of independent signals [38, 14, 32, 19, 1, 2]. Although exhibiting low computational costs, those algorithms, being based on gradient descent techniques, have similar drawbacks. The most relevant is that there is no guarantee of convergence towards each minima of the CM cost function, especially if the number P of sources is not known a priori. These very important problems are solved by the analytical constant modulus algorithm (ACMA) [45]. In fact, ACMA has the following properties. In the noiseless case: **(i)** for a number of sources $P \leq N$, $K > P^2$ array snapshots are sufficient to compute $\boldsymbol{H}$ and $\boldsymbol{S}$ exactly via an eigenvalue problem; **(ii)** for $K > P^2$, it is possible to detect the number of CM signals present in the K

array snapshots $\boldsymbol{X}$. In the noisy observations case: **(iii)** robustness in finding $\boldsymbol{S}$ can be achieved. We introduce ACMA in this subsection. We focus on the most relevant issues that support the derivation of the algorithm.

The CM factorization problem. We start from the data model in eq. (12), where $\boldsymbol{H}$ and $\boldsymbol{S}$ are assumed full rank, and model the constant modulus signal property as

$$\boldsymbol{X} = \boldsymbol{H}\boldsymbol{S}, \quad |S_{ij}| = 1. \tag{19}$$

With generality, it is assumed that only $\delta \leq P$ are CM signals. If the factorization in (19) is unique[1], then the CM factorization problem can be formulated in an equivalent way: given the $(N \times K)$ data matrix $\boldsymbol{X}$ of rank P, find δ and the $(\delta \times N)$ matrix $\boldsymbol{W}$, such that

$$\boldsymbol{W}\boldsymbol{X} = \boldsymbol{S}_\delta, \quad |(S_\delta)_{ij}| = 1, \tag{20}$$

where the $\delta \times K$ matrix $\boldsymbol{S}_\delta$ is full rank, and $\delta \leq P$ is as large as possible. Let $\mathrm{row}(\boldsymbol{X})$ denote the subspace spanned by the rows of $\boldsymbol{X}$, and define the set of CM signals

$$\mathcal{CM} = \{\boldsymbol{S} \mid |S_{ij}| = 1 \ , \forall \ i, j\}. \tag{21}$$

To solve the CM factorization problem, we have therefore to find the rows $\boldsymbol{w}$ of $\boldsymbol{W}$ such that $\boldsymbol{w}\boldsymbol{X} = \boldsymbol{s}$, $\boldsymbol{s} \in \mathrm{row}(\boldsymbol{X})$ being one of the linearly independent signals in $\mathcal{CM}$. This is equivalent to finding all linearly independent signals $\boldsymbol{s}$ that satisfy

$$\begin{aligned} &(A) \quad \boldsymbol{s} \in \mathrm{row}(\boldsymbol{X}) \\ &(B) \quad \boldsymbol{s} \in \mathcal{CM}. \end{aligned}$$

The Gerchberg–Saxton Algorithm (GSA). The signals $\boldsymbol{s}$, satisfying (A) and (B), can be found iteratively using an adequate alternating projections based algorithm. Suppose that $\boldsymbol{y} = \boldsymbol{w}^{(i)}\boldsymbol{X}$ is a signal in the row span of $\boldsymbol{X}$ at iteration i. To guarantee that $\boldsymbol{y}$ belongs to $\mathcal{CM}$, consider the non-linear projector onto $\mathcal{CM}$

$$\boldsymbol{P}_{\mathcal{CM}}(y) = \left[\frac{(\boldsymbol{y})_1}{|(\boldsymbol{y})_1|}, \ldots, \frac{(\boldsymbol{y})_K}{|(\boldsymbol{y})_K|}\right].$$

The iteration is then

$$\boldsymbol{w}^{(i+1)} = \left[\boldsymbol{P}_{\mathcal{CM}}(\boldsymbol{w}^{(i)}\boldsymbol{X})\right]\boldsymbol{X}^\dagger. \tag{22}$$

For each signal of interest, GSA is initialized with a different random choice of $\boldsymbol{s}$.

The problem with this type of solution is that the finiteness of the data sets can preclude global convergence and may introduce spurious local minima. Reference [45] reformulates the problem in such a way that an analytical solution is provided.

[1] As in the last subsection, *uniqueness* is assumed up to an admissible transformation matrix.

Equivalent formulation. Consider the singular value decomposition of $\boldsymbol{X}$

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}: \ \boldsymbol{U} \in \mathbb{C}^{N\times N}, \ \boldsymbol{\Sigma} \in \mathbb{R}^{N\times K}, \ \boldsymbol{V} \in \mathbb{C}^{K\times K}, \tag{23}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are unitary matrices containing the singular vectors of $\boldsymbol{X}$, and $\boldsymbol{\Sigma}$ is a real diagonal matrix with non-negative entries (in descending order), the singular values of $\boldsymbol{X}$. If P is the number of signals, then $\text{rank}(\boldsymbol{X}) = P$ is the number of non-zero singular values of $\boldsymbol{X}$. Thus, the first P rows of $\boldsymbol{V}$ form an orthonormal basis of the space $\text{row}(\boldsymbol{X})$, which we collect in $\widehat{\boldsymbol{V}} \in \mathbb{C}^{P\times K}$. Condition (A) can then be rewritten as

$$(A'): \ \boldsymbol{s} \in \text{row}(\boldsymbol{X}) \quad \Leftrightarrow \quad \boldsymbol{s} = \boldsymbol{w}\widehat{\boldsymbol{V}}, \quad \widehat{\boldsymbol{V}} \in \mathbb{C}^{P\times K}. \tag{24}$$

Notice that in this equivalent condition (A'), the row vector $\boldsymbol{w}$ has only P elements as a result of using the orthonormal basis in $\widehat{\boldsymbol{V}}$ instead of the data matrix $\boldsymbol{X}$.

We write the $P \times K$ matrix $\widehat{\boldsymbol{V}}$ as

$$\widehat{\boldsymbol{V}} = [\boldsymbol{v}_1 \cdots \boldsymbol{v}_k \cdots \boldsymbol{v}_K]$$

where $\boldsymbol{v}_k \in \mathbb{C}^P$ is the kth-column of $\widehat{\boldsymbol{V}}$. Define

$$\boldsymbol{P}_k = \boldsymbol{v}_k\boldsymbol{v}_k^H \ \in \mathbb{C}^{P\times P}, \quad k = 1, \ldots, K.$$

Condition (B) becomes

$$(B): \ \boldsymbol{s} = [\ (\boldsymbol{s})_1 \cdots (\boldsymbol{s})_K] \in \mathcal{CM} \ \Leftrightarrow \left[\ |(\boldsymbol{s})_1|^2 \cdots |(\boldsymbol{s})_K|^2\right] = [1 \cdots 1] \Leftrightarrow \begin{cases} \boldsymbol{w}\boldsymbol{v}_1\boldsymbol{v}_1^H\boldsymbol{w}^H = 1 \\ \vdots \\ \boldsymbol{w}\boldsymbol{v}_K\boldsymbol{v}_K^H\boldsymbol{w}^H = 1, \end{cases}$$

which is also equivalent to

$$(B'): \ \boldsymbol{s} \in \mathcal{CM} \quad \Leftrightarrow \quad \boldsymbol{w}\boldsymbol{P}_k\boldsymbol{w}^H = 1, \quad k = 1, \ldots, K. \tag{25}$$

From condition (B') it follows that to solve the CM factorization problem, we must find the solutions $\boldsymbol{w}$ to the K quadratic eqs. (25). We rewrite these conditions using the Kronecker product[2] $\otimes$ and the following property of $\otimes$, [16]. For $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, $\boldsymbol{B}_1$, and $\boldsymbol{B}_2$ matrices with appropriate dimensions

$$(\boldsymbol{A}_1 \cdot \boldsymbol{A}_2) \otimes (\boldsymbol{B}_1 \cdot \boldsymbol{B}_2) = (\boldsymbol{A}_1 \otimes \boldsymbol{B}_1) \cdot (\boldsymbol{A}_2 \otimes \boldsymbol{B}_2) \tag{26}$$

Then,

$$\boldsymbol{w}\boldsymbol{P}_k\boldsymbol{w}^H = \boldsymbol{w}\boldsymbol{v}_k\boldsymbol{v}_k^H\boldsymbol{w}^H = \boldsymbol{v}_k^H\boldsymbol{w}^H\boldsymbol{w}\boldsymbol{v}_k = \boldsymbol{v}_k^H\boldsymbol{w}^H\boldsymbol{v}_k^T\boldsymbol{w}^T.$$

These equalities follow because $\boldsymbol{w}$ is a $(1 \times P)$ row vector, $\boldsymbol{v}_k$ is a $(P \times 1)$ column vector, and $\boldsymbol{w}\boldsymbol{v}_k$ is a scalar. Since the product of two scalars commutes and equals its Kronecker product, we obtain further

$$\boldsymbol{w}\boldsymbol{P}_k\boldsymbol{w}^H = \left(\boldsymbol{v}_k^H\boldsymbol{w}^H\right) \otimes \left(\boldsymbol{v}_k^T\boldsymbol{w}^T\right) = \left(\boldsymbol{v}_k^H \otimes \boldsymbol{v}_k^T\right) \cdot \left(\boldsymbol{w}^H \otimes \boldsymbol{w}^T\right), \qquad k = 1, \cdots, K.$$

[2] The Kronecker product of two matrices $\boldsymbol{A} = [a_{ij}]$ and $\boldsymbol{B}$ is $\boldsymbol{A} \otimes \boldsymbol{B} = [a_{ij}\boldsymbol{B}]$.

The last equality follows from the property (26). Recall that $\boldsymbol{A}^T \otimes \boldsymbol{B}^T = (\boldsymbol{A} \otimes \boldsymbol{B})^T$ and define

$$\boldsymbol{y} = \boldsymbol{w}^H \otimes \boldsymbol{w}^T \in \mathbb{C}^{P^2 \times 1} \qquad \text{and} \qquad \boldsymbol{P} = \begin{bmatrix} \boldsymbol{p}_1 \\ \vdots \\ \boldsymbol{p}_K \end{bmatrix} = \begin{bmatrix} [\boldsymbol{v}_1^* \otimes \boldsymbol{v}_1]^T \\ \vdots \\ [\boldsymbol{v}_K^* \otimes \boldsymbol{v}_K]^T \end{bmatrix} \in \mathbb{C}^{K \times P^2}. \tag{27}$$

Using these, the K equations in condition (B') in eq. (25) are organized in matrix form

$$\boldsymbol{P}\boldsymbol{y} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}. \tag{28}$$

This is shown in [45]. The CM factorization problem translates now into finding all linearly independent vector solutions to eq. (28). Clearly, for each solution $\boldsymbol{w}$ of eq. (28), $\boldsymbol{s} = \boldsymbol{w}\widehat{\boldsymbol{V}}$ is the corresponding CM signal.

The solution space of (28) can be generally written as an affine space $\boldsymbol{y} = \boldsymbol{y}_0 + \alpha_1 \boldsymbol{y}_1 + \cdots + \alpha_\ell \boldsymbol{y}_\ell$, where $\boldsymbol{y}_0$ is a particular solution of (28), and $\{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_\ell\}$ is a basis of the kernel of $\boldsymbol{P}$. To work with a fully linear solution space, a linear transformation can be used. Consider a $(K \times K)$ unitary matrix $\boldsymbol{Q}$ such that

$$\boldsymbol{Q} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} K^{1/2} \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{29}$$

For instance, $\boldsymbol{Q}$ can be chosen as the Discrete Fourier Transform (DFT) or a Householder transformation,

$$\boldsymbol{Q} = \boldsymbol{I}_K - 2\frac{\boldsymbol{q}\boldsymbol{q}^H}{\boldsymbol{q}^H\boldsymbol{q}}, \qquad \boldsymbol{q} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} - \begin{bmatrix} K^{1/2} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{30}$$

which applies to $\boldsymbol{P}$ as follows:

$$\boldsymbol{Q}\boldsymbol{P} \doteq \begin{bmatrix} \widehat{\boldsymbol{p}}_1 \\ \widehat{\boldsymbol{P}} \end{bmatrix}, \qquad \begin{cases} \widehat{\boldsymbol{p}}_1 \in \mathbb{C}^{1 \times P^2} \\ \widehat{\boldsymbol{P}} \in \mathbb{C}^{(K-1) \times P^2} \end{cases}. \tag{31}$$

Then

$$\boldsymbol{P}\boldsymbol{y} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \iff \begin{cases} \text{(i)} \quad \widehat{\boldsymbol{p}}_1 \boldsymbol{y} = K^{1/2} \\ \text{(ii)} \quad \widehat{\boldsymbol{P}}\boldsymbol{y} = \boldsymbol{0}, \end{cases} \tag{32}$$

and all the linear independent nonzero solutions $\boldsymbol{y}$ of (28) also satisfy

$$\begin{aligned} \widehat{\boldsymbol{P}}\boldsymbol{y} &= \boldsymbol{0} \\ \boldsymbol{y} &= \boldsymbol{w}^H \otimes \boldsymbol{w}^T. \end{aligned} \tag{33}$$

Now, let $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{\widehat{\delta}}\}$ be a basis of the kernel of $\widehat{\boldsymbol{P}}$, where $\widehat{\delta}$ is its respective dimension. Therefore, any solution $\boldsymbol{y}$ of $\widehat{\boldsymbol{P}}\boldsymbol{y} = \boldsymbol{0}$ can be written as $\boldsymbol{y} = \alpha_1\boldsymbol{y}_1 + \ldots + \alpha_{\widehat{\delta}}\boldsymbol{y}_{\widehat{\delta}}$. Using another property of the Kronecker product, [16], namely that for a column vector $\boldsymbol{a}$ and a row vector $\boldsymbol{b}$, both of the same dimension, $\mathrm{vec}\,(\boldsymbol{a}\cdot\boldsymbol{b}) = \boldsymbol{b}^T \otimes \boldsymbol{a}$, the condition $\boldsymbol{y} = \boldsymbol{w}^H \otimes \boldsymbol{w}^T$ can also be written as $\boldsymbol{Y} = \boldsymbol{w}^T\boldsymbol{w}^*$, where $\boldsymbol{Y} = \mathrm{vec}^{-1}(\boldsymbol{y})$. Then,

$$\alpha_1\boldsymbol{y}_1 + \cdots + \alpha_{\widehat{\delta}}\boldsymbol{y}_{\widehat{\delta}} = \boldsymbol{w}^H \otimes \boldsymbol{w}^T \quad \Leftrightarrow \quad \alpha_1\boldsymbol{Y}_1 + \cdots + \alpha_{\widehat{\delta}}\boldsymbol{Y}_{\widehat{\delta}} = \boldsymbol{w}^T\boldsymbol{w}^*,$$

i.e., the conditions (33) are rewritten as a linear combination of the matrices $\{\boldsymbol{Y}_i\}_{i=1}^{\widehat{\delta}}$, $\boldsymbol{Y}_i = \mathrm{vec}^{-1}(\boldsymbol{y}_i)$, such that this linear combination is a rank one Hermitian matrix, hence factorizable as $\boldsymbol{w}^T\boldsymbol{w}^*$. Linear independent solutions $\boldsymbol{w}$ lead to linear independent solutions $\boldsymbol{y}$ that in turn lead to linear independent parameter vectors $[\alpha_1 \cdots \alpha_{\widehat{\delta}}]$.

The CM problem is reformulated then as follows.

Let $\boldsymbol{X}$ be the data matrix from which the set of $(P \times P)$ matrices $\{\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_{\widehat{\delta}}\}$ are derived as described before. The CM problem is then equivalent to the determination of all independent nonzero parameter vectors $[\alpha_1 \cdots \alpha_{\widehat{\delta}}]$ such that

$$\alpha_1\boldsymbol{Y}_1 + \cdots + \alpha_{\widehat{\delta}}\boldsymbol{Y}_{\widehat{\delta}} = \boldsymbol{w}^T\boldsymbol{w}^*. \tag{34}$$

For each solution $\boldsymbol{w}$, $\|\boldsymbol{w}\| = K^{1/2}$, the vector $\boldsymbol{s} = \boldsymbol{w}\widehat{\boldsymbol{V}}$ is a CM signal in $\boldsymbol{X}$.

Solution of the noiseless CM problem. The exact solution of the noiseless CM factorization problem is obtained in two steps: **(step 1)** computation of the number δ of CM signals; and **(step 2)** computation of the δ row vectors $\boldsymbol{w}$ in (33).

Step1. Reference [45] shows that, for $K > P^2$, the dimension $\widehat{\delta}$ of the kernel of $\widehat{\boldsymbol{P}}$ equals, in general, the number δ of CM signals present in $\boldsymbol{X}$. The only situations where $\widehat{\delta} > \delta$ occurs is when specific phase relations exist between the signals that are present. This can be the case with BPSK and MSK signals sampled at the signal rate. These degeneracies disappear when these signals are fractionally sampled. Therefore, for almost all the cases of interest, the number of CM signals present in $\boldsymbol{X}$ is obtained by computing the dimension of the kernel of $\widehat{\boldsymbol{P}}$.

Step 2. Assume $K > P^2$ and that $\widehat{\delta} = \delta$, $\widehat{\delta}$ being the dimension of the kernel of $\widehat{\boldsymbol{P}}$. Let the δ linear independent solutions to the CM problem be $\boldsymbol{w}_1^H \otimes \boldsymbol{w}_1^T, \cdots, \boldsymbol{w}_\delta^H \otimes \boldsymbol{w}_\delta^T$. They are a basis for the kernel space of $\widehat{\boldsymbol{P}}$. It can be shown, [45], that each of the matrices $\boldsymbol{Y}_i$ can be expressed as a linear combination

of these basis vectors $\boldsymbol{w}_j^T \boldsymbol{w}_j^*$. Writing these δ independent linear combinations for the δ matrices $\boldsymbol{Y}_i$ leads to the following.

Collect the $\boldsymbol{w}_i$ in the matrix

$$\boldsymbol{W} = \left[\boldsymbol{w}_1^T \cdots \boldsymbol{w}_\delta^T\right]^T. \tag{35}$$

Then, the simultaneous independent linear combinations of the $\boldsymbol{Y}_i$ shows that the CM factorization problem is equivalent to finding the $(\delta \times P)$ matrix $\boldsymbol{W}$ with full rank–δ such that

$$\begin{aligned} \boldsymbol{Y}_1 &= \boldsymbol{W}^T \boldsymbol{\Lambda}_1 \boldsymbol{W}^* \\ \boldsymbol{Y}_2 &= \boldsymbol{W}^T \boldsymbol{\Lambda}_2 \boldsymbol{W}^* \\ &\cdots \\ \boldsymbol{Y}_\delta &= \boldsymbol{W}^T \boldsymbol{\Lambda}_\delta \boldsymbol{W}^* \\ \boldsymbol{\Lambda}_1, \ldots \boldsymbol{\Lambda}_\delta &\in \mathbb{C}^{P \times P}, \text{ diagonal matrices.} \end{aligned} \tag{36}$$

This is a simultaneous diagonalization problem that can be solved using the *Super–Generalized Schur Decomposition*, see [45] for the details.

The CM factorization problem with noisy observations. Here we consider the noisy observations model (11),

$$\boldsymbol{X} = \boldsymbol{H}\boldsymbol{S} + \boldsymbol{N}.$$

In this case, it is not possible to obtain an exact CM factorization. However, the CM factorization problem of noisy data $\boldsymbol{X}$ can be formulated as an optimization problem in the context of an appropriate CM metric, such as

$$\text{dist}(\boldsymbol{s}, \mathcal{CM}) = \sum_{k=1}^{K} \left(|(\boldsymbol{s})_k|^2 - 1 \right)^2. \tag{37}$$

Therefore, we must find δ signals $\boldsymbol{s}$ that are minimizers of

$$\min \left\{ \text{dist}(\boldsymbol{s}, \mathcal{CM}) | \boldsymbol{s} \in \widehat{\text{row}(\boldsymbol{X})} \right\}, \tag{38}$$

$\widehat{\text{row}(\boldsymbol{X})}$ being the estimated row span of $\boldsymbol{S}$, i.e., the principal row span of $\boldsymbol{X}$ determined by an SVD, see eq. (23). Letting P be the number of signals, then the principal row span of $\boldsymbol{X}$ is as before determined by the matrix $\widehat{\boldsymbol{V}}$, which collects the P orthonormal P rows of $\boldsymbol{V}$ corresponding to the P largest singular values of $\boldsymbol{X}$. Like in the noiseless situation, the matrices $\boldsymbol{P}$ and $\widehat{\boldsymbol{P}}$ can be constructed from $\widehat{\boldsymbol{V}}$.

It can be shown, [45], that the CM problem with noise is solved by finding all linearly independent minimizers $\boldsymbol{y}$ of $\left\| \widehat{\boldsymbol{P}} \boldsymbol{y} \right\|^2$, subject to

$$\boldsymbol{y} = \boldsymbol{w}^H \otimes \boldsymbol{w}^T, \quad \|\boldsymbol{w}\| = K^{1/2}.$$

As in the noiseless case, the solution to this optimization problem is based on the set of δ matrices $\boldsymbol{Y} = \mathrm{vec}^{-1}(\boldsymbol{y})$. Linear combinations of these matrices should result in matrices close to rank–1 matrices of the form

$$\alpha_1 \boldsymbol{Y}_1 + \cdots + \alpha_\delta \boldsymbol{Y}_\delta = \boldsymbol{Y} \simeq \boldsymbol{w}^T \boldsymbol{w}^*. \tag{39}$$

Again, the problem of finding all the δ independent parameter vectors $[\alpha_1 \cdots \alpha_\delta]$ can be solved based on a super–generalized Schur decomposition. The procedure departs again from eqs. (36) and starts with a QR and RQ factorizations of $\boldsymbol{W}^H$ and $\boldsymbol{W}$, respectively. Let $\boldsymbol{W}^T = \boldsymbol{Q}^H \boldsymbol{R}'$ and $\boldsymbol{W}^* = \boldsymbol{R}'' \boldsymbol{Z}^H$ where $\boldsymbol{Q}$ and $\boldsymbol{Z}$ are unitary and $\boldsymbol{R}'$ and $\boldsymbol{R}''$ are upper triangular. Then pre-multiplying the ith-equation in (36) on the left by the $(P \times P)$ matrix $\boldsymbol{Q}$ and on the right by the $(P \times P)$ matrix $\boldsymbol{Z}$

$$\boldsymbol{Q} \boldsymbol{Y}_i \boldsymbol{Z} = \boldsymbol{R}_i, \quad i = 1, \ldots, \delta, \tag{40}$$

where $\left\{ \boldsymbol{R}_i \in \mathbb{C}^{P \times P} \right\}_{i=1}^{\delta}$, $\boldsymbol{R}_i = \boldsymbol{R}' \boldsymbol{\Lambda}_i \boldsymbol{R}''$, are upper triangular matrices. It is possible to show that a parameter vector $[\alpha_1 \cdots \alpha_\delta]$ satisfies condition (39) only if

$$\alpha_1 \boldsymbol{R}_1 + \cdots + \alpha_\delta \boldsymbol{R}_\delta \qquad \text{is rank 1.}$$

Given the decomposition of the $\boldsymbol{R}_i$, we get the equivalent condition

$$\alpha_1 \boldsymbol{\Lambda}_1 + \cdots + \alpha_\delta \boldsymbol{\Lambda}_\delta \qquad \text{is rank 1.}$$

Because the $\boldsymbol{\Lambda}_i$ are diagonal, this linear combination is diagonal. In other words, only one entry of this diagonal matrix, say entry i, is nonzero. Set this entry to one,

$$\alpha_1^i \left(\boldsymbol{\Lambda}_1\right)_{ii} + \cdots + \alpha_\delta^i \left(\boldsymbol{\Lambda}_\delta\right)_{ii} = 1 \qquad i = 1, \cdots, \delta.$$

Collecting these δ equations in matrix format, let $\boldsymbol{A}$ be a $\delta \times \delta$ matrix whose ith row is $\left[\alpha_1^i, \cdots, \alpha_\delta^i\right]$ and let $\boldsymbol{\Lambda}$ the matrix whose ith row is the diagonal of $\boldsymbol{\Lambda}_i$. Then

$$\boldsymbol{A} \boldsymbol{\Lambda} = \boldsymbol{I}$$

and the rows of $\boldsymbol{\Lambda}^{-1}$ are the desired independent vectors $[\alpha_1 \cdots \alpha_\delta]$.

In [45], it is shown that in fact one does not need to perform the factorization of the $\boldsymbol{R}_i$, since an equivalent result is obtained from the rows of $\boldsymbol{A}$:

$$\boldsymbol{A} = \boldsymbol{R}^{-1}, \quad \boldsymbol{R} = \begin{bmatrix} (\boldsymbol{R}_1)_{11} & \cdots & (\boldsymbol{R}_1)_{\delta\delta} \\ \vdots & \vdots & \vdots \\ (\boldsymbol{R}_\delta)_{11} & \cdots & (\boldsymbol{R}_\delta)_{\delta\delta} \end{bmatrix}. \tag{41}$$

Once these δ independent parameter vectors $[\alpha_1 \cdots \alpha_\delta]$ that verify (39) are found, each row vector $\boldsymbol{w}$ can then be estimated as the singular vector corresponding to the largest singular value of each $\boldsymbol{Y}$.

The simultaneous upper triangularization problem specified in (40) is solved using an extended $\boldsymbol{QZ}$ iteration described in [45]. The problem with this iteration is that there is no proof of convergence, although in practice this can be achieved in a few number of iterations. We summarize the algorithm next.

Analytical Constant Modulus Algorithm

1. Estimation of row($\boldsymbol{X}$):
   (a) Compute an SVD of $\boldsymbol{X}$, eq.(23);
   (b) Estimate the number $P = \text{rank}(\boldsymbol{X})$ of signals from $\boldsymbol{\Sigma}$ in eq.(23);
   (c) Define $\widehat{\boldsymbol{V}}$ (first $\boldsymbol{P}$ rows of $\boldsymbol{V}$ in eq.(23)).
2. Estimation of δ = dimension of the kernel of $\widehat{\boldsymbol{P}}$:
   (a) Construct the $((K-1) \times P^2)$ matrix $\widehat{\boldsymbol{P}}$ from $\widehat{\boldsymbol{V}} = [\boldsymbol{v}_1 \cdots \boldsymbol{v}_K]$, using eqs.(27-31);
   (b) Compute an SVD of $\widehat{\boldsymbol{P}}$: $\widehat{\boldsymbol{P}} = \boldsymbol{U_P \Sigma_P V_P}$;
   (c) Estimate δ from $\boldsymbol{\Sigma_P}$;
   (d) Define $[\boldsymbol{y}_1 \cdots \boldsymbol{y}_\delta]$, the last δ columns of $\boldsymbol{V_P}$.
3. Solve the simultaneous upper triangularization problem in eq.(40):
   (a) Define $\boldsymbol{Y}_i = \text{vec}^{-1}(\boldsymbol{y}_i),\ i = 1, \ldots, \delta$;
   (b) Find $\boldsymbol{R}_i,\ i = 1, \ldots, \delta$, in eq.(40);
   (c) Find all vectors $[\alpha_{i1} \cdots \alpha_{i\delta}],\ i = 1, \ldots, \delta$, from the rows of $\boldsymbol{A}$ in eq.(41);
   (d) Compute $\widehat{\boldsymbol{Y}}_i = \alpha_{i1}\boldsymbol{Y}_1 + \cdots + \alpha_{i\delta}\boldsymbol{Y}_\delta,\ i = 1, \ldots, \delta$.
4. Recover the CM signals. For each $\widehat{\boldsymbol{Y}}_i$:
   (a) Compute $\boldsymbol{w}_i$ such that $\widehat{\boldsymbol{Y}}_i \simeq \boldsymbol{w}_i^T \boldsymbol{w}_i^*$;
   (b) Scale: make $\|\boldsymbol{w}_i\| = K^{1/2}$;
   (c) $\boldsymbol{s}_i = \boldsymbol{w}_i \widehat{\boldsymbol{V}},\ i = 1, \ldots, \delta$, are the rows of $\boldsymbol{S}$ that are CM signals.

The ACM algorithm presents several interesting properties: **(i)** it is deterministic, which means that the minima of the cost function are obtained analytically; and **(ii)** it is robust with respect to both the length of the data sequences, and to the presence of weak noise. However, due to the SVDs involved in the algorithm, it presents a high computational complexity when compared with other CM algorithms, e.g., the GSA in eq. (22). Even if the SVDs are computed using efficient algorithms, the complexity of the ACMA is approximately $9P^4K + 36N^2K$ flops, while the GSA takes $80PNK + 8N^2K$ flops. Although ACMA is an elegant analytic solution to the approximate factorization problem of the noisy observation

data, the ACMA relies on an iterative procedure, the extended $\boldsymbol{QZ}$ iteration, which is not guaranteed to converge to the desired solution.

2.1.3 Closed form solution based on linear coding

Here, we describe a closed–form solution to the factorization problem that we have been addressing. In contradistinction with ACMA, this solution does not rely on an iterative procedure and has a much lower computational complexity. The main idea is to encode the source data using a simple linear coding scheme, which maintains the data rate and does not increase the necessary transmission bandwidth. Moreover, the resulting closed–form solution enables reconstructing the information data without requiring first the identification of the channel mixing matrix.

Consider the noiseless observations model

$$\boldsymbol{X} = \boldsymbol{H}\boldsymbol{Z}, \tag{42}$$

where each row of $\boldsymbol{Z}$, $(\boldsymbol{z})_p$, $p = 1, \ldots, P$, represents the encoded data transmitted by user p,

$$(\boldsymbol{z})_p \leftarrow \text{linear encoding}\left((\boldsymbol{s})_p\right), \; p = 1, \ldots, P, \tag{43}$$

and $(\boldsymbol{s})_p$, $p = 1, \ldots, P$, is the binary information data generated by user p. In [42], the encoding scheme is determined by complex diagonal matrices $\boldsymbol{D}_p$, $p = 1, \ldots, P$, with symbol entries in some complex alphabet $\mathcal{E} \subset \mathbb{C}$, so that

$$(\boldsymbol{z})_p = (\boldsymbol{s})_p \boldsymbol{D}_p \; \in \mathbb{C}^{(1\times K)}, \; p = 1, \ldots, P. \tag{44}$$

Source separation. Given $\boldsymbol{X}$ in (42), and assuming (44), the objective is to obtain all the binary signals $(\boldsymbol{s})_p$, $p = 1, \ldots, P$, transmitted by the sources. The algorithm starts with the SVD of the data matrix

$$\begin{aligned} \boldsymbol{X} &= \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^H \\ &= [\boldsymbol{U}_s \; \boldsymbol{U}_n] \begin{bmatrix} \boldsymbol{\Sigma}_s & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_s^H \\ \boldsymbol{V}_n^H \end{bmatrix}. \end{aligned} \tag{45}$$

Assuming that $\boldsymbol{H}$ and $\boldsymbol{Z}$ are, respectively, full column and full row rank–P matrices, then $\boldsymbol{Z}$ spans the same row space as $\boldsymbol{V}_s^H$. Since $\boldsymbol{V}_s \perp \boldsymbol{V}_n$, i.e., $\boldsymbol{V}_s^H \boldsymbol{V}_n = \boldsymbol{0}$, then $\boldsymbol{Z}\boldsymbol{V}_n = \boldsymbol{0}$. Defining

$$\boldsymbol{V}_{n_p} = \boldsymbol{D}_p \boldsymbol{V}_n, \qquad p = 1, \ldots, P, \tag{46}$$

then

$$(\boldsymbol{s})_p \boldsymbol{V}_{n_p} = \boldsymbol{0}, \qquad p = 1, \ldots, P. \tag{47}$$

This constrains the encoding matrix $\boldsymbol{D}_p$. Taking into account that the information data is real,

$$(\boldsymbol{s})_p \left[\begin{array}{cc} \mathcal{R}(\boldsymbol{V}_{n_p}) & \mathcal{I}(\boldsymbol{V}_{\boldsymbol{n_p}}) \end{array} \right] = \boldsymbol{0}, \qquad p = 1, \ldots, P \tag{48}$$

which doubles the constraints.

Selection of the coding matrices. Given the definition (46) and eq.(48), we conclude that each coding matrix $\boldsymbol{D}_p$, $p = 1, \ldots, P$, must be selected so as to guarantee that the left null space of $\left[\begin{array}{cc} \mathcal{R}(\boldsymbol{V}_{n_p}) & \mathcal{I}(\boldsymbol{V}_{\boldsymbol{n_p}}) \end{array} \right]$ is 1–dimensional. As shown in [42], this is achievable if and only if the mild condition $K \geq 2P - 1$ is verified.

When noise is present, $\boldsymbol{X} = \boldsymbol{H}\boldsymbol{Z} + \boldsymbol{N}$, the matrix $\boldsymbol{V}_n$ and all the $\boldsymbol{V}_{n_p}$'s (see eq. (46)) are noise dependent, and eq. (47) is not verified exactly. This means that the selection of each coding matrix $\boldsymbol{D}_p$, $p = 1, \ldots, P$, must follow some statistical criterion that approximates the 1–dimensional condition on the left kernel of $\left[\begin{array}{cc} \mathcal{R}(\boldsymbol{V}_{n_p}) & \mathcal{I}(\boldsymbol{V}_{\boldsymbol{n_p}}) \end{array} \right]$, $p = 1, \ldots, P$.

The linear coding based closed–form solution performs worst than both the ACMA and the ILSP algorithm. This results from its relative simplicity, since it avoids the identification of the channel mixing matrix. The payoff is that it being a closed–form algorithm, the linear coding based solution results in low computational complexity when compared with other alternatives.

2.2 SUBSPACE METHOD FOR ISI CANCELLATION

Consider the linear convolutive mixture model developed in section 1 for the case of P independent sources, which are transmitted through different multipath channels, all of them having the same time length L. Thus

$$\boldsymbol{x}(k) = \boldsymbol{H}\,\boldsymbol{s}(k) + \boldsymbol{n}(k), \qquad k = 1, \ldots, K, \tag{49}$$

where, as before, $\boldsymbol{x}(k)$ and $\boldsymbol{n}(k)$ are the N–dimensional vectors of array data and noise, respectively. Here,

$$\boldsymbol{s}(k) = \left[\boldsymbol{s}_1^T(k) \cdots \boldsymbol{s}_P^T(k)\right]^T,$$

with

$$\boldsymbol{s}_p(k) = \left[s_p(k)\ s_p(k-1) \cdots s_p(k-L+1)\right]^T,$$

and $\boldsymbol{H}$ is the $(N \times M)$, $M = PL$, channel mixture matrix. For reasons that will become clearer shortly, we rearrange the first term on the right–hand side of (49) through a permutation. Let

$$\boldsymbol{L}_L^{PL} \tag{50}$$

be the permutation matrix of dimension $M = PL$ and stride L. We will refer to $\boldsymbol{L}_L^{PL}$ as the *multipath* permutation. The multipath permutation applied on the left to the vector $\boldsymbol{s}(k)$ reshufles its components by reading in the first component, then the component $L+1$, then $2L+1$, and so on, i.e., the vector $\boldsymbol{s}(k)$ is rearranged into a new vector

$$\widetilde{\boldsymbol{s}}(k) = \left[\widetilde{\boldsymbol{s}}_0^T(k)\ \widetilde{\boldsymbol{s}}_1^T(k) \cdots \widetilde{\boldsymbol{s}}_{L-1}^T(k)\right]^T, \tag{51}$$

where

$$\widetilde{\boldsymbol{s}}_l(k) = [s_1(k-l)\ s_2(k-l)\cdots s_P(k-l)]^T, \qquad l = 0, 1, \ldots, L-1. \tag{52}$$

It is easy to show that the inverse of the multipath permutation $\boldsymbol{L}_L^{PL}$ is the permutation $\boldsymbol{L}_P^{PL}$, referred to as the *channel multipath* permutation. Then, inserting $\boldsymbol{L}_P^{PL}\boldsymbol{L}_L^{PL}$ in between $\boldsymbol{H}\boldsymbol{s}(k)$ in the first term of the data model (49), we can write it as

$$\boldsymbol{x}(k) = \widetilde{\boldsymbol{H}}\widetilde{\boldsymbol{s}}(k) + \boldsymbol{n}(k),\ k = 1, \ldots, K; \tag{53}$$

the matrix $\widetilde{\boldsymbol{H}} = \boldsymbol{H}\boldsymbol{L}_P^{PL}$, i.e., it is $\boldsymbol{H}$ up to a column permutation. Collecting as in section 1 all the K array snapshots in an $(N \times K)$ matrix,

$$\boldsymbol{X} = \widetilde{\boldsymbol{H}}\widetilde{\boldsymbol{S}} + \boldsymbol{N}, \tag{54}$$

where $\widetilde{\boldsymbol{S}}$ is a block Toeplitz matrix as can be verified by direct substitution. In fact,

$$\widetilde{\boldsymbol{S}} = \begin{bmatrix} \widetilde{\boldsymbol{s}}_0(1) & \ddots & \widetilde{\boldsymbol{s}}_0(K-1) & \widetilde{\boldsymbol{s}}_0(K) \\ \ddots & \ddots & \ddots & \widetilde{\boldsymbol{s}}_1(K) \\ \widetilde{\boldsymbol{s}}_{L-2}(1) & \widetilde{\boldsymbol{s}}_{L-2}(2) & \ddots & \ddots \\ \widetilde{\boldsymbol{s}}_{L-1}(1) & \widetilde{\boldsymbol{s}}_{L-1}(2) & \ddots & \widetilde{\boldsymbol{s}}_{L-1}(K) \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_1 & \ddots & \boldsymbol{s}_{K-1} & \boldsymbol{s}_K \\ \ddots & \ddots & \ddots & \boldsymbol{s}_{K-1} \\ \boldsymbol{s}_{-L+3} & \boldsymbol{s}_{-L+4} & \ddots & \ddots \\ \boldsymbol{s}_{-L+2} & \boldsymbol{s}_{-L+3} & \ddots & \boldsymbol{s}_{K-L+1} \end{bmatrix} \tag{55}$$

where, for $l = 0, \cdots, L-1$ and $k = 1, \cdots, K$, the (l,k)-block entries of the matrix on the right of eq. (55) are $\widetilde{\boldsymbol{s}}_l(k) = \boldsymbol{s}_{k-l}$, as can be verified. By the Toeplitz condition, $\boldsymbol{s}_{k-l} = \widetilde{\boldsymbol{s}}_l(k) = \widetilde{\boldsymbol{s}}_{l+n}(k+n)$ for any integer n.

It is the block Toeplitz structure of the signal matrix $\widetilde{\boldsymbol{S}}$ that is exploited by the signal subspace method. The signal subspace method essentially contributes to canceling the ISI effect for each source. Basicaly, the channel convolutive mixture matrix is translated into an equivalent instantaneous mixture of P independent sources. These are then separated using one of the available algorithms, such as ILSP, ILSE, or ACMA.

In the following, we assume that $\widetilde{\boldsymbol{H}}$ and $\widetilde{\boldsymbol{S}}$ are matrices with full column and full row rank–M, respectively. These conditions imply that the row and column spans of $\boldsymbol{X}$ equal the row span of $\widetilde{\boldsymbol{S}}$ and the column span of $\widetilde{\boldsymbol{H}}$, respectively. Formally,

$$\begin{aligned} \widetilde{\boldsymbol{H}} \text{ full column rank} &\implies \text{row}(\boldsymbol{X}) = \text{row}(\widetilde{\boldsymbol{S}}) \\ \widetilde{\boldsymbol{S}} \text{ full row rank} &\implies \text{col }(\boldsymbol{X}) = \text{col }(\widetilde{\boldsymbol{H}}). \end{aligned} \tag{56}$$

The factorization $\boldsymbol{X} = \widetilde{\boldsymbol{H}}\widetilde{\boldsymbol{S}}$ can be achieved by finding either $\widetilde{\boldsymbol{S}}$ or $\widetilde{\boldsymbol{H}}$ with a specified row or column span, respectively; in fact, those row or column spans of $\boldsymbol{X}$, as expressed in the necessary conditions (56). Here, we will follow the first strategy, where the block Toeplitz structure of $\widetilde{\boldsymbol{S}}$ is exploited.

Estimation of the row span of $\widetilde{\boldsymbol{S}}$. As discussed before, our assumptions guarantee that row($\widetilde{\boldsymbol{S}}$) can be estimated from row($\boldsymbol{X}$). The SVD of $\boldsymbol{X}$ produces the factorization $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^H$, where $\boldsymbol{U}$ and $\boldsymbol{V}$

are unitary matrices, and $\boldsymbol{\Sigma}$ is a diagonal matrix whose entries are the singular values (in descending order) of $\boldsymbol{X}$. In the absence of noise, $\boldsymbol{X}$ is rank–M, and Σ has exactly M non–zero diagonal entries. In this case, we can write $\boldsymbol{X} = \widehat{\boldsymbol{U}}\widehat{\boldsymbol{\Sigma}}\widehat{\boldsymbol{V}}^H$, where the entries of the $(M \times M)$ diagonal matrix $\widehat{\boldsymbol{\Sigma}}$ are the non–zero entries of $\boldsymbol{\Sigma}$, $\widehat{\boldsymbol{U}}$ consists of the first M columns of $\boldsymbol{U}$, and $\widehat{\boldsymbol{V}}^H$ consists of the first M rows of $\boldsymbol{V}^H$. Therefore

$$\begin{aligned} \text{row}(\widehat{\boldsymbol{V}}^H) &= \text{row}(\widetilde{\boldsymbol{S}}) \\ \text{col }(\widehat{\boldsymbol{U}}) &= \text{col }(\widetilde{\boldsymbol{H}}). \end{aligned} \tag{57}$$

When noise is present, the rank of $\boldsymbol{X}$ is estimated as the number of singular values that are above the noise level. To increase the robustness against noise, this detection problem is solved based on the eigenvalues of $\boldsymbol{X}\boldsymbol{X}^H = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^H$. For white noise with covariance matrix $\sigma^2\boldsymbol{I}_N$, and for large enough K, the diagonal matrix $\boldsymbol{\Lambda}$ will have $N - M$ smallest eigenvalues $\lambda_{\text{noise}} \simeq K\sigma^2$ and M largest eigenvalues $\lambda_m \simeq (\widehat{\Sigma})^2_{mm} + K\sigma^2$, $m = 1, \ldots, M$.

Forcing the Toeplitz structure of $\widetilde{\boldsymbol{S}}$. Now that we have a basis $\widehat{\boldsymbol{V}}^H$ to span the row space of $\widetilde{\boldsymbol{S}}$, we find a description of $\widetilde{\boldsymbol{S}}$ that has a block Toeplitz structure with L block rows, as in (55). Following [44], this is done using a technique denoted *row span intersections*.

The equivalence of the row spaces in (57) means that each row of $\widetilde{\boldsymbol{S}} \in \text{row}\left(\boldsymbol{V}^H\right)$. We work from this condition and reexpress it in an alternative way.

Collect the distinct block entries of $\widetilde{\boldsymbol{S}}$ into the block row matrix

$$\boldsymbol{\mathcal{S}} = \left[\underbrace{\boldsymbol{s}_{-L+2}\ \boldsymbol{s}_{-L+3} \cdots \boldsymbol{s}_1 \cdots \boldsymbol{s}_{K-L+1}}_{\text{last row block of } \widetilde{\boldsymbol{S}}}\ \boldsymbol{s}_{K-L+2} \cdots \boldsymbol{s}_K \right], \tag{58}$$

in other words, $\boldsymbol{\mathcal{S}}$ is the generator of the block Toeplitz matrix $\widetilde{\boldsymbol{S}}$. For example, the last row block of $\widetilde{\boldsymbol{S}}$ is explicitly shown in eq. (58). By sliding one entry to the right the under brace we get the second to last row block of $\widetilde{\boldsymbol{S}}$. Finally, the last K block entries of $\boldsymbol{\mathcal{S}}$ are the first row of $\boldsymbol{\mathcal{S}}$. Define the following shifts of the row space of $\boldsymbol{X}$, row$\boldsymbol{X}$, suitably embedded with zeros

$$\widehat{\boldsymbol{V}}^{(l)^H} \doteq \begin{bmatrix} \boldsymbol{0} & \widehat{\boldsymbol{V}}^H & \boldsymbol{0} \\ \boldsymbol{I}_{l-1} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_{L-l} \end{bmatrix}, \qquad l = 1, \ldots, L. \tag{59}$$

For example,

$$\widehat{\boldsymbol{V}}^{(1)^H} = \begin{bmatrix} \widehat{\boldsymbol{V}}^H & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{L-1} \end{bmatrix}, \ \widehat{\boldsymbol{V}}^{(2)^H} = \begin{bmatrix} \boldsymbol{0} & \widehat{\boldsymbol{V}}^H & \boldsymbol{0} \\ 1 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_{L-2} \end{bmatrix}, \text{ and } \widehat{\boldsymbol{V}}^{(L)^H} = \begin{bmatrix} \boldsymbol{0} & \widehat{\boldsymbol{V}}^H \\ \boldsymbol{I}_{L-1} & \boldsymbol{0} \end{bmatrix}.$$

Then, it can be shown that each block row of $\widetilde{\boldsymbol{S}}$ is in row$\boldsymbol{X}$ if the following L conditions are satisfied

$$\boldsymbol{\mathcal{S}} \in \text{row}(\widehat{\boldsymbol{V}}^{(l)^H}), \qquad l = 1, \cdots, L. \tag{60}$$

In words, $\boldsymbol{\mathcal{S}}$ is in the subspace intersection of the row spaces $\mathrm{row}(\widehat{\boldsymbol{V}}^{(1)^H}), \cdots, \mathrm{row}(\widehat{\boldsymbol{V}}^{(L)^H})$. We interpret this condition. Consider $l = 1$.

$$\boldsymbol{\mathcal{S}} = [\boldsymbol{s}_{-L+2}\ \boldsymbol{s}_{-L+3} \cdots \boldsymbol{s}_1 \cdots \boldsymbol{s}_{K-L+1}\ \boldsymbol{s}_{K-L+2} \cdots \boldsymbol{s}_K] \in \mathrm{row}(\widehat{\boldsymbol{V}}^{(1)^H}).$$

This condition places no restriction on the last $(L-1)$ block entries of $\boldsymbol{\mathcal{S}}$ and states that the (block) row of its first K block entries

$$[\boldsymbol{s}_{-L+2}\ \boldsymbol{s}_{-L+3} \cdots \boldsymbol{s}_1 \cdots \boldsymbol{s}_{K-L+1}] \in \mathrm{row}(\widehat{\boldsymbol{V}}^{(1)^H}),$$

which states exactly that the last row of $\widetilde{\boldsymbol{S}}$ is in $\mathrm{row}(\widehat{\boldsymbol{V}}^{(1)^H})$. The remaining $(L-1)$ conditions in eq. (60) are similarly interpreted.

We now consider the problem of determining the intersection subspace of the L subspaces $\mathrm{row}(\widehat{\boldsymbol{V}}^{(l)^H})$. This intersection subspace can be computed using De Morgan's Laws through the complement of the direct sum of the complements of each subspace. This direct computation of the intersection subspace turns out to be innefficient for this problem of source separation (see the discussion in the Appendix of [44]). An alternative is to form the matrix that stacks the orthogonal generators of each of the subspaces $\mathrm{row}(\widehat{\boldsymbol{V}}^{(l)^H})$. One way of doing this is to compute the singular value decomposition (SVD) of the matrix that stacks the matrices $\widehat{\boldsymbol{V}}^{(l)^H}$

$$\begin{bmatrix} \widehat{\boldsymbol{V}}^{(1)^H} \\ \vdots \\ \widehat{\boldsymbol{V}}^{(L)^H} \end{bmatrix}. \tag{61}$$

Since we are not interested on the left singular vectors of (61), it can be shown that it is equivalent to compute instead the singular values and the right singular vectors of

$$\boldsymbol{V}_{(L)} = \begin{bmatrix} \boxed{\widehat{\boldsymbol{V}}^H} & & \boldsymbol{0} \\ & \boxed{\widehat{\boldsymbol{V}}^H} & \\ \ddots & & \ddots \\ \boldsymbol{0} & & \boxed{\widehat{\boldsymbol{V}}^H} \\ \boxed{\boldsymbol{J}_1} & & \boldsymbol{0} \\ \boldsymbol{0} & & \boxed{\boldsymbol{J}_2} \end{bmatrix}, \tag{62}$$

which includes L one entry–shifted consecutive copies of $\widehat{\boldsymbol{V}}^H$, and the matrices

$$\boldsymbol{J}_1 = \begin{bmatrix} \sqrt{L-1} & & \boldsymbol{0} \\ & \ddots & \\ & & \sqrt{2} \\ \boldsymbol{0} & & 1 \end{bmatrix} \quad \text{and} \quad \boldsymbol{J}_2 = \begin{bmatrix} 1 & & \boldsymbol{0} \\ & \sqrt{2} & \\ & & \ddots \\ \boldsymbol{0} & & \sqrt{L-1} \end{bmatrix}. \tag{63}$$

These matrices account for the linear independent rows of the identity matrices stacked in (61).

The intersection of the row spans of $\widehat{\boldsymbol{V}}^{(1)^H}, \ldots, \widehat{\boldsymbol{V}}^{(L)^H}$, which determines the ISI free P information data signals, has a basis $\boldsymbol{Y}$ given by the right singular vectors corresponding to the largest singular values of $\boldsymbol{V}_{(L)}$. When there is no noise, it is shown in [44] that $\boldsymbol{V}_{(L)}$ has precisely P largest singular values equal to $\sqrt{L}$, while the smallest are approximately equal to $\sqrt{L-1}$. So, we determine the interception subspace by computing the L right singular vectors of $\boldsymbol{V}_{(L)}$ corresponding to the largest singular value $\sqrt{L}$. Notice that, for large L, this ISI filtering may be a very delicate issue.

At this point, we have a basis $\boldsymbol{Y}$ of the signal space where the users information data signals lie. Let $\boldsymbol{\mathcal{S}}_P$ be the matrix whose P rows are these data signals. Then, we can write

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{\mathcal{S}}_P, \tag{64}$$

where $\boldsymbol{A}$ is some arbitrary matrix. Naturally, the ILSP or the ILSE algorithms can now be used to find the factorization (64), subject to the elements in $\boldsymbol{\mathcal{S}}_P$ being in a finite alphabet.

In this paragraph, we have introduced the main ideas involved in the subspace method for blind ISI cancellation and source separation. For more details on the algorithm see [44].

3 SECOND-ORDER STATISTICS METHODS

We study blind multi-user signal processing techniques that exploit the information conveyed by the second order statistics (SOS) of the received data.

In subsection 3.1, the SOS are used to prewhiten the observed data set. Roughly, the observed data vectors are projected on a dimension-reduced space, where the channel matrix, although still unknown, is unitary, i.e., a rotation matrix. From this algebraic property, together with other source characteristics, we derive computationally attractive source separation algorithms and/or intersymbol interference rejection techniques.

In subsection 3.2, the SOS are used to obtain analytical solutions for the blind channel identifiability problem. The approaches to be discussed rely on some level of pre-processing at the transmitter to attain these closed-form solutions. Basically, pre-filters located at the transmitters insert a sufficiently rich structure into the correlation matrices, in order to assist the receiver in its blind source decoupling task.

3.1 ITERATIVE METHODS: DATA PREWHITENING

Consider the noisy linear convolutive mixture of P sources developed in section 1

$$\boldsymbol{x}(k) = \sum_{p=1}^{P} \boldsymbol{H}_p \boldsymbol{s}_p(k) + \boldsymbol{n}(k). \tag{65}$$

Hereafter, for the sake of clarity, we assume that the P channels have equal time lengths, $L_1 = L_2 = \cdots = L_P = L$, i.e., the sources are exposed to the same degree of intersymbol interference (ISI). We

rewrite as previously done eq. (65) as

$$\boldsymbol{x}(k) = \boldsymbol{H}\boldsymbol{s}(k) + \boldsymbol{n}(k), \tag{66}$$

where the channel matrix and the sources signal are

$$\begin{aligned} \boldsymbol{H} &= [\,\boldsymbol{H}_1\,\boldsymbol{H}_2\,\cdots\,\boldsymbol{H}_P\,], \\ \boldsymbol{s}(k) &= \left[\,\boldsymbol{s}_1(k)^T\,\boldsymbol{s}_2(k)^T\,\cdots\,\boldsymbol{s}_P(k)^T\,\right]^T. \end{aligned}$$

The assumptions on the data model in eq. (66) are:

(A1) The $N \times M$ channel matrix $\boldsymbol{H}$ is full column rank. The dimensions are $M = PL$ and $N \geq M$;

(A2) The sources $s_p(k)$ denote zero-mean, uncorrelated wide-sense stationary processes. Moreover, the sources emit uncorrelated data samples with unit power. This entails no loss of generality, as multiplicative constants are absorbed in $\boldsymbol{H}$. Thus,

$$r_{p,q}(k,l) = \mathrm{E}\,\{s_p(k)s_q(l)^*\} = \delta(p-q, k-l),$$

where $\delta(n,m)$ denotes the Kronecker delta: $\delta(n,m) = 1$ if $(n,m) = (0,0)$, $\delta(n,m) = 0$ if $(n,m) \neq (0,0)$. In matrix notation, the autocorrelation of $\boldsymbol{s}(k)$, is

$$\boldsymbol{R}_{\boldsymbol{s}}(k) = \mathrm{E}\left\{\boldsymbol{s}(l)\boldsymbol{s}(l-k)^H\right\} = \boldsymbol{I}_M\delta(k);$$

(A3) For simplicity, the noise $\boldsymbol{n}(k)$ is a zero-mean spatio-temporal white Gaussian process with known variance σ^2, i.e.,

$$\boldsymbol{R}_{\boldsymbol{n}}(k) = \mathrm{E}\left\{\boldsymbol{n}(l)\boldsymbol{n}(l-k)^H\right\} = \sigma^2\boldsymbol{I}_N\delta(k).$$

The noise $\boldsymbol{n}(k)$ is independent of the source signals $s_p(k)$.

Data prewhitening converts the unknown channel matrix $\boldsymbol{H}$ in (66) into a (still unknown) unitary matrix. The unitary structure simplifies signal processing problems such as co-channel source resolution, see paragraph 3.1.1, and multi-user echo suppression, see paragraph 3.1.2. Also, a geometric interpretation of the operation of the algorithms becomes readily available.

Let $\boldsymbol{R}_{\boldsymbol{x}}(k)$ be the correlation matrix of the observations $\boldsymbol{x}(k)$ at lag $k \in \mathbb{Z}$, defined as $\boldsymbol{R}_{\boldsymbol{s}}(k)$ and $\boldsymbol{R}_{\boldsymbol{w}}(k)$,

$$\boldsymbol{R}_{\boldsymbol{x}}(k) = \mathrm{E}\left\{\boldsymbol{x}(l)\boldsymbol{x}(l-k)^H\right\}.$$

Data prewhitening may be accomplished as follows. Given assumptions (A2) and (A3),

$$\begin{aligned} \boldsymbol{R}_{\boldsymbol{x}}(0) &= \boldsymbol{H}\boldsymbol{R}_{\boldsymbol{s}}(0)\boldsymbol{H}^H + \boldsymbol{R}_{\boldsymbol{w}}(0) \\ &= \boldsymbol{H}\boldsymbol{H}^H + \sigma^2\boldsymbol{I}_N. \end{aligned} \tag{67}$$

Denote the eigenvalue decomposition (EVD) of $\boldsymbol{R}_{\boldsymbol{x}}(0)$ by

$$\boldsymbol{R}_{\boldsymbol{x}}(0) = \boldsymbol{U}\boldsymbol{\Sigma}^2\boldsymbol{U}^H$$

where the $N \times N$ unitary matrix $\boldsymbol{U}$ and the $N \times N$ diagonal matrix $\boldsymbol{\Sigma}^2$ are

$$\begin{aligned} \boldsymbol{U} &= [\boldsymbol{U}_1\, \boldsymbol{U}_2] \\ \boldsymbol{\Sigma}^2 &= \text{diag}\left(\boldsymbol{\Sigma}_1^2 + \sigma^2\boldsymbol{I}_M, \sigma^2\boldsymbol{I}_{N-M}\right). \end{aligned}$$

The block $\boldsymbol{U}_1$ is $N \times M$, and $\boldsymbol{\Sigma}_1$ is $M \times M$, diagonal, with positive diagonal entries in descending order. As seen, $\boldsymbol{\Sigma}_1$ is available from the EVD of $\boldsymbol{R}_{\boldsymbol{x}}(0)$ by subtracting σ^2 from every diagonal entry in the upper-left $M \times M$ block of $\boldsymbol{\Sigma}^2$ and taking square roots to the resulting entries. From

$$\begin{aligned} \boldsymbol{R}_{\boldsymbol{x}}(0) &= \boldsymbol{U}_1\boldsymbol{\Sigma}_1^2\boldsymbol{U}_1^H + \sigma^2\boldsymbol{I}_N \\ &= \widehat{\boldsymbol{H}}\widehat{\boldsymbol{H}}^H, \end{aligned} \tag{68}$$

where we defined $\widehat{\boldsymbol{H}} \equiv \boldsymbol{U}_1\boldsymbol{\Sigma}_1$. From eq. (68) and eq. (67), it follows

$$\widehat{\boldsymbol{H}}\widehat{\boldsymbol{H}}^H = \boldsymbol{H}\boldsymbol{H}^H.$$

This, given assumption (A1) and standard algebra, implies in turn that

$$\widehat{\boldsymbol{H}} = \boldsymbol{H}\boldsymbol{Q}^H$$

for some (unknown) unitary $M \times M$ matrix $\boldsymbol{Q}$. Thus,

$$\widehat{\boldsymbol{H}}^\dagger = \boldsymbol{Q}\boldsymbol{H}^\dagger$$

and whitened data samples are obtained as

$$\boldsymbol{y}(k) \equiv \widehat{\boldsymbol{H}}^\dagger\boldsymbol{x}(k) = \boldsymbol{Q}\boldsymbol{s}(k) + \boldsymbol{w}(k), \tag{69}$$

where $\boldsymbol{w}(k) \equiv \widehat{\boldsymbol{H}}^\dagger\boldsymbol{n}(k)$. Notice that $\boldsymbol{y}(k) \in \mathbb{C}^M$, whereas $\boldsymbol{x}(k) \in \mathbb{C}^N$. Thus, recalling assumption (A1), since $M \leq N$, the equivalent projected data samples $\boldsymbol{y}(k)$ live in a dimension-reduced data space. This implies that the computational complexity of the algorithms in the remaining processing pipeline is reduced, as they tipically depend on the dimensionality of the data samples.

3.1.1 Instantaneous mixtures

In this paragraph, we outline two iterative source separation algorithms that exploit the unitary structure of the channel matrix $\boldsymbol{Q}$ in the prewhiten data samples $\boldsymbol{y}(k)$ in (69): **(i)** the hypercube algorithm [17, 18]; and **(ii)** the least-square (LS) constellation derotator [46]. Here, we restrict ourselves to instantaneous mixtures, i.e., $L_p = 1$ in eq. (65), or, equivalently, $\boldsymbol{Q}$ is a $P \times P$ matrix in eq. (69). An SOS-based technique that converts convolutive mixtures into instantaneous ones is discussed in paragraph 3.1.2.

Also, for clarity, we examine only the case of binary sources, i.e., the information signal $s_p(k)$ consists of independent identically distributed (i.i.d.) data bits taken from the binary alphabet $\mathcal{A} = \{\pm 1\}$. Moreover, without loss of generality, all data are assumed to be real, i.e., $\boldsymbol{x}(k) \in \mathbb{R}^N$, $\boldsymbol{y}(k) \in \mathbb{R}^P$, $\boldsymbol{Q} \in \mathbb{R}^{P \times P}$, and so on.

Hypercube algorithm. The hypercube algorithm is a sequential source separation technique that recursively extracts all the P transmitted data streams, one at a time. Once a user's signal is estimated, its contribution is removed from the observed mixture – the dimension of the problem is deflated by one – and the algorithm re-starts searching for another signal. Suppose K is the available number of data vectors $\boldsymbol{y}(k)$. Then, collecting the data vectors in a $P \times K$ data matrix $\boldsymbol{Y}$, we get

$$\begin{aligned} \boldsymbol{Y} &= \left[\begin{array}{cccc} \boldsymbol{y}(1) & \boldsymbol{y}(2) & \cdots & \boldsymbol{y}(K) \end{array} \right] \\ \boldsymbol{Y} &= \boldsymbol{Q}\boldsymbol{S} + \boldsymbol{W}, \end{aligned} \tag{70}$$

where the $P \times K$ signal and noise matrices $\boldsymbol{S}$ and $\boldsymbol{W}$ follow similar definitions as in eq. (70). For noiseless samples, $\boldsymbol{W} = \boldsymbol{0}$, and, for a certain condition on $\boldsymbol{S}$, it turns out that the columns of $\boldsymbol{Q}$ are the unique maximizers, up to a $\pm$ sign, of a certain function formulated over the observed data matrix.

Recall that for a generic vector $\boldsymbol{x} = [\, x_1 \; x_2 \; \cdots \; x_n \,]^T \in \mathbb{R}^n$ its l_p norm is given by

$$l_p(\boldsymbol{x}) = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}.$$

Theorem 3.1 *Let $\boldsymbol{Y} = \boldsymbol{Q}\boldsymbol{S}$. Suppose that $\boldsymbol{S} : P \times 2^P$ contains all 2^P combinations of ± 1's among its columns, and let $\boldsymbol{Q}$ be unitary. If $\boldsymbol{\alpha} \in \mathbb{R}^P$ is a global maximizer of $f(\boldsymbol{\alpha}) = l_1(\boldsymbol{Y}^T\boldsymbol{\alpha})$, subject to $l_2(\boldsymbol{\alpha}) = 1$, then $\boldsymbol{\alpha} = \pm \boldsymbol{q}_p$, where $\boldsymbol{q}_p$ denotes some column of $\boldsymbol{Q}$* $\square$.

The objective function in *Theorem* 3.1 is given by

$$f(\boldsymbol{\alpha}) = \sum_{k=1}^{K} \left| \boldsymbol{y}(k)^T \boldsymbol{\alpha} \right|.$$

Given a global maximizer of f, say $\widehat{\boldsymbol{\alpha}}$, and since $\boldsymbol{Q}$ is unitary, then

$$\widehat{\boldsymbol{s}}^T = \operatorname{sign}(\widehat{\boldsymbol{\alpha}}^T \boldsymbol{Y})$$

denotes a row of $\boldsymbol{S}$, i.e., some transmitted binary stream.

Maximization of the objective function $f(\boldsymbol{\alpha})$, subject to the constraint $\|\boldsymbol{\alpha}\| = 1$, is achieved through a standard gradient search tecnhique, as follows. Since for arbitrary $x \in \mathbb{R}$, $|x| = \operatorname{sign}(x)x$, then

$$f(\boldsymbol{\alpha}) = \boldsymbol{\alpha}^T \boldsymbol{\beta}, \qquad \text{where} \qquad \boldsymbol{\beta} = \sum_{k=1}^{K} \operatorname{sign}(\boldsymbol{\alpha}^T \boldsymbol{y}(k)) \boldsymbol{y}(k).$$

Thus, assuming that $\boldsymbol{\alpha}^T \boldsymbol{y}(k) \neq 0$, for all $k = 1, \ldots, K$,

$$\boldsymbol{\nabla} f(\boldsymbol{\alpha}) = \boldsymbol{\beta}.$$

The locally convergent gradient search subroutine is given below.

1. Set $i = 0$ and choose $\boldsymbol{\alpha}_0$
   a) $i = i + 1$
   b) $\widehat{\boldsymbol{s}}_i^T = \text{sign}(\boldsymbol{\alpha}_{i-1}^T \boldsymbol{Y})$
   c) $\boldsymbol{\beta} = \boldsymbol{Y}\widehat{\boldsymbol{s}}_i / \left\|\boldsymbol{Y}\widehat{\boldsymbol{s}}_i\right\|$
   d) $\boldsymbol{\alpha}_i = \boldsymbol{\alpha}_{i-1} + \left[\boldsymbol{I} - \boldsymbol{\alpha}_{i-1}\boldsymbol{\alpha}_{i-1}^T\right]\boldsymbol{\beta}$
   e) $\boldsymbol{\alpha}_i = \boldsymbol{\alpha}_i / \left\|\boldsymbol{\alpha}_i\right\|$
2. until $\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_{i-1} = \boldsymbol{0}$
3. $\widehat{\boldsymbol{s}}^T = \text{sign}(\boldsymbol{\alpha}^T \boldsymbol{Y})$.

Step 1.d) projects the gradient $\boldsymbol{\beta}$ onto the tangent space of the constraint set $\|\boldsymbol{\alpha}_{i-1}\| = 1$, and then moves in the direction of this projected gradient (in order to maximize f). Step 1.e) returns to the feasible set.

In practice, there are departures from the ideal conditions assumed so far: the whitened data samples $\boldsymbol{Y} = \boldsymbol{Q}\boldsymbol{S} + \boldsymbol{W}$ are corrupted by additive colored noise $\boldsymbol{W}$; the signal matrix $\boldsymbol{S}$ possesses some correlation among its rows, i.e., the rows of $\boldsymbol{S}$ are not necessarily orthogonal. The hypercube algorithm takes into account these non-ideal conditions: after the kth step, all the prior information is exploited to correct the estimates produced: $\boldsymbol{\alpha}_k$; $\widehat{\boldsymbol{s}}_k^T$; and $\widehat{\boldsymbol{q}}_k = \boldsymbol{Y}\widehat{\boldsymbol{s}}_k^T / \left\|\widehat{\boldsymbol{s}}_k\right\|^2$, an estimate of the kth column of $\boldsymbol{Q}$. The first k columns of $\boldsymbol{Q}$, i.e., $\widehat{\boldsymbol{Q}}_k = [\,\widehat{\boldsymbol{q}}_1 \cdots \widehat{\boldsymbol{q}}_k\,]$ are re-estimated as $\widehat{\boldsymbol{Q}}_k = \boldsymbol{Y}\widehat{\boldsymbol{S}}_k^{\dagger}$, where

$$\widehat{\boldsymbol{S}}_k = \begin{bmatrix} \widehat{\boldsymbol{s}}_1^T \\ \vdots \\ \widehat{\boldsymbol{s}}_k^T \end{bmatrix}$$

contains the currently extracted k user binary signals; the pseudo-inverse handles the correlation among the rows of $\boldsymbol{S}$.

The estimate of $\boldsymbol{\alpha}_k$ is improved as follows. Ideally, $\boldsymbol{\alpha}_k$ denotes the kth column of $\boldsymbol{Q}$, and, being orthogonal to the other columns of the unitary $\boldsymbol{Q}$, it spans the null space of

$$\widetilde{\boldsymbol{Q}}_k^T = \left[\,\boldsymbol{q}_1 \cdots \boldsymbol{q}_{k-1}\,\boldsymbol{q}_{k+1} \cdots \boldsymbol{q}_P\,\right]^T,$$

or, equivalently, the null space of

$$\widetilde{\boldsymbol{R}}_k = \widetilde{\boldsymbol{Q}}_k \widetilde{\boldsymbol{Q}}_k^T.$$

On the other hand, $\widetilde{\boldsymbol{R}}_k$ is the denoised correlation matrix

$$\widetilde{\boldsymbol{R}}_k = \text{E}\left\{\widetilde{\boldsymbol{y}}(k)\widetilde{\boldsymbol{y}}(k)^T\right\} - \boldsymbol{R}_{\boldsymbol{w}}, \qquad \text{with} \qquad \widetilde{\boldsymbol{y}}(k) = \boldsymbol{y}(k) - \boldsymbol{q}_k s(k),$$

where the noise correlation matrix $\boldsymbol{R}_{\boldsymbol{w}}$ is assumed known. Thus, $\boldsymbol{\alpha}_k$ is re-estimated as the eigenvector associated with the smallest eigenvalue of

$$\frac{1}{K}\widetilde{\boldsymbol{Y}}\widetilde{\boldsymbol{Y}}^T - \boldsymbol{R}_{\boldsymbol{w}}, \qquad \text{where} \qquad \widetilde{\boldsymbol{Y}} = \boldsymbol{Y} - \widetilde{\boldsymbol{q}}_k \widehat{\boldsymbol{s}}_k^T.$$

Reference [17] offers an alternative geometric interpretation of the re-estimation of $\boldsymbol{\alpha}_k$.

The next step deflates the dimension of the problem by one. This is achieved by applying an oblique projector to the observed data $\boldsymbol{Y}$. The range of the oblique projector is set to the null space of

$$\widehat{\boldsymbol{Q}}_k^{-1} \equiv \begin{bmatrix} \widehat{\boldsymbol{\alpha}}_1^T \\ \vdots \\ \widehat{\boldsymbol{\alpha}}_k^T \end{bmatrix},$$

and its null space to the range of $\widehat{\boldsymbol{Q}}_k$, in order to reject the extracted signals and keep the unextracted ones untouched. The final step consists in reducing the dimensionality of the oblique projector. Details to implement this overall projector, say $\boldsymbol{\Pi}$, can be found in [5]. The complete hypercube algorithm is listed below.

1. Initialization: set $\boldsymbol{Y}_1 = \boldsymbol{Y}$, $\widehat{\boldsymbol{Q}}_0^{-1} = \emptyset$, $\widehat{\boldsymbol{S}}_0 = \emptyset$
2. for $k = 1$ to $P - 1$
   a) call gradient search subroutine with input $\boldsymbol{Y}_k$, return $\widehat{\boldsymbol{s}}_k^T$
   b) $\widehat{\boldsymbol{S}}_k = \begin{bmatrix} \widehat{\boldsymbol{S}}_{k-1} \\ \widehat{\boldsymbol{s}}_k^T \end{bmatrix}$
   c) $\widehat{\boldsymbol{Q}}_k = \boldsymbol{Y}\widehat{\boldsymbol{S}}_k^{\dagger}$
   d) Re-estimate $\boldsymbol{\alpha}_k$ as the eigenvector associated with the smallest eigenvalue of $\widetilde{\boldsymbol{Y}}\widetilde{\boldsymbol{Y}}^T/K - \boldsymbol{R}_{\boldsymbol{w}}$
   e) $\widehat{\boldsymbol{Q}}_k^{-1} = \begin{bmatrix} \widehat{\boldsymbol{Q}}_{k-1}^{-1} \\ \boldsymbol{\alpha}_k^T \end{bmatrix}$
   f) Project the observed data $\boldsymbol{Y}(P \times K)$: $\boldsymbol{Y}_{k+1} = \boldsymbol{\Pi}\boldsymbol{Y}((P-k) \times K)$
3. $\widehat{\boldsymbol{s}}_P^T = \text{sign}(\boldsymbol{Y}_P)$
4. $\widehat{\boldsymbol{S}}_P = \begin{bmatrix} \widehat{\boldsymbol{S}}_{P-1} \\ \widehat{\boldsymbol{s}}_P^T \end{bmatrix}$, $\widehat{\boldsymbol{Q}}_P = \boldsymbol{Y}\widehat{\boldsymbol{S}}_P^{\dagger}$.

Least-squares constellation derotator. The whitened data samples obey the model

$$\boldsymbol{y}(k) = \boldsymbol{Q}\boldsymbol{s}(k) + \boldsymbol{w}(k),$$

where the $P \times P$ matrix $\boldsymbol{Q}$ is orthogonal and $\boldsymbol{s}(k)$ belongs to the P-dimensional binary constellation

$$\mathcal{B}^P = \mathcal{B} \times \cdots \times \mathcal{B}, \qquad \text{where} \qquad \mathcal{B} = \{\pm 1\}.$$

The cardinality of $\mathcal{B}^P$ is 2^P. Thus, geometrically, the samples $\boldsymbol{y}(k)$ are obtained by applying the rotation $\boldsymbol{Q}$ to the hypercube whose vertices are given by

$$\mathcal{H} = \mathcal{B}^P = \left\{(\pm 1, \ldots, \pm 1)^T\right\},$$

and adding noise $\boldsymbol{w}(k)$. In other words, the observations $\boldsymbol{y}(k)$, see Figure 2, form clusters around the vertices of the rotated hypercube

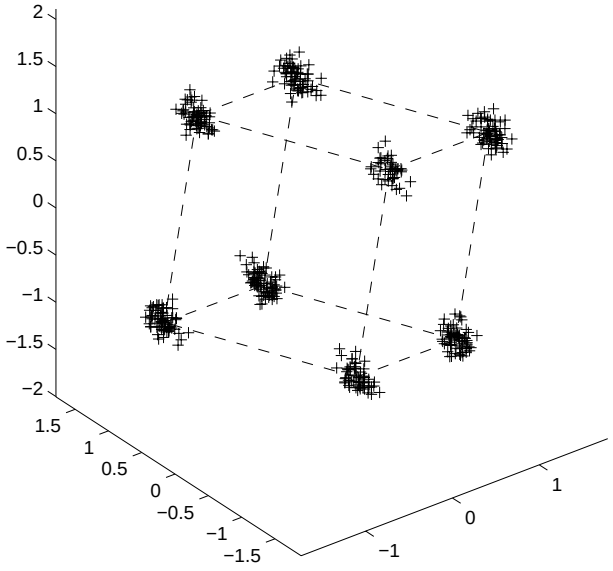

Figure 2: Geometric interpretation of $\boldsymbol{y}(k)$: clusters centered about the vertices of $\mathcal{H}_{\boldsymbol{Q}}$

This geometrical interpretation motivates the following approach for identifying the unknown rotation $\boldsymbol{Q}$: find the orthogonal matrix $\widehat{\boldsymbol{Q}}$ that "best" derotates the observed samples $\boldsymbol{y}(k)$, more specifically, that minimizes the least-squares distance of the derotated samples to the reference constellation hypercube $\mathcal{H}$. Thus, if $\mathrm{dist}(\boldsymbol{x},\mathcal{H}) = \min_{\boldsymbol{b}\in\mathcal{H}} \|\boldsymbol{x}-\boldsymbol{b}\|$ and $\mathcal{U} = \{P \times P \text{ orthogonal matrices}\}$, we have

$$\widehat{\boldsymbol{Q}} = \underset{\boldsymbol{U}\,\in\,\mathcal{U}}{\arg\min} \sum_{k=1}^{K} \mathrm{dist}^2\left(\boldsymbol{U}^T\boldsymbol{y}(k),\mathcal{H}\right). \tag{71}$$

The function $\mathrm{dist}(\boldsymbol{x},\mathcal{H})$ denotes the distance from the point $\boldsymbol{x} \in \mathbb{R}^P$ to the hypercube $\mathcal{H}$. Letting the vertex of $\mathcal{H}$ closest to $\boldsymbol{x}$ be

$$\boldsymbol{b}(\boldsymbol{x}) = \underset{\boldsymbol{b}\,\in\,\mathcal{H}}{\arg\min}\ \|\boldsymbol{x}-\boldsymbol{b}\|,$$

and using the fact that $\boldsymbol{U}$ is orthogonal, eq. (71) can be rewritten as

$$\widehat{\boldsymbol{Q}} = \underset{\boldsymbol{U}\,\in\,\mathcal{U}}{\arg\min} \sum_{k=1}^{K} \left\|\boldsymbol{y}(k) - \boldsymbol{U}\boldsymbol{b}\left(\boldsymbol{U}^T\boldsymbol{y}(k)\right)\right\|^2. \tag{72}$$

The formulation in eq. (72) does not admit a closed-form solution, but the alternative minimization

$$\left(\widehat{\boldsymbol{Q}}, \widehat{\boldsymbol{S}}\right) = \underset{(\boldsymbol{U}, \boldsymbol{B}) \in \mathcal{U} \times \mathcal{B}}{\arg\min} \sum_{k=1}^{K} \|\boldsymbol{y}(k) - \boldsymbol{U}\boldsymbol{b}(k)\|^2 , \tag{73}$$

where $\boldsymbol{B} = [\,\boldsymbol{b}(1) \cdots \boldsymbol{b}(K)\,]$ and $\mathcal{B} = \{P \times K \text{ binary matrices}\}$, yields, in the first component of the ordered pair, a solution to eq. (72). The minimization in eq. (73) can be handled iteratively by separating the variables $\boldsymbol{U}$ and $\boldsymbol{B}$. Namely, given the nth iteration $(\boldsymbol{U}_n, \boldsymbol{B}_n)$, we let

$$\boldsymbol{U}_{n+1} = \underset{\boldsymbol{U} \in \mathcal{U}}{\arg\min} \|\boldsymbol{Y} - \boldsymbol{U}\boldsymbol{B}_n\|^2 \tag{74}$$

$$\boldsymbol{B}_{n+1} = \underset{\boldsymbol{B} \in \mathcal{B}}{\arg\min} \|\boldsymbol{Y} - \boldsymbol{U}_{n+1}\boldsymbol{B}\|^2 . \tag{75}$$

After algebraic manipulations, the subproblem in (74) is equivalent to

$$\boldsymbol{U}_{n+1} = \underset{\boldsymbol{U} \in \mathcal{U}}{\arg\max} \operatorname{tr}\left(\boldsymbol{B}_n \boldsymbol{Y}^T \boldsymbol{U}\right),$$

whose solution is given by the transpose of the polar factor of the $P \times P$ matrix $\boldsymbol{B}_n\boldsymbol{Y}^T$, see [13]; more specifically, if

$$\boldsymbol{B}_n\boldsymbol{Y}^T = \boldsymbol{V}_1\boldsymbol{\Sigma}\boldsymbol{V}_2^T$$

denotes a singular value decomposition of $\boldsymbol{B}_n\boldsymbol{Y}^T$, then

$$\boldsymbol{U}_{n+1} = \boldsymbol{V}_2\boldsymbol{V}_1^T.$$

With respect to the optimization problem expressed in eq. (75), we notice that the minimization can be carried out column-by-column of $\boldsymbol{B}$, each column being found by exhaustive search in the constellation $\mathcal{B}^P$, i.e., the kth column of $\boldsymbol{B}_{n+1}$ is given by

$$\boldsymbol{b}_{n+1}(k) = \underset{\boldsymbol{b} \in \mathcal{B}^P}{\arg\min} \|\boldsymbol{y}(k) - \boldsymbol{U}_n\boldsymbol{b}\|^2 .$$

3.1.2 ISI cancellation methods

We rework the data model of eq. (69), by regrouping terms with the same delay in the multipath replicas. This is similar to what was done with eq. (51) and eq. (52). Recall the multipath permutation matrix $\boldsymbol{L}_L^M$, $M = PL$, with stride L introduced in eq. (50) and its inverse the channel multipath permutation $\boldsymbol{L}_P^M$. Define the permuted vector $\widetilde{\boldsymbol{s}}_l(k)$ and the matrix $\boldsymbol{U}$ obtained by a column permutation of the unitary

matrix $\boldsymbol{Q}$, by applying the channel multipath permutation $\boldsymbol{L}_P^M$ on the right. Then eq. (69) is rewritten

$$\begin{aligned}
\boldsymbol{y}(k) &= \sum_{l=0}^{L-1} \boldsymbol{U}_l \widetilde{\boldsymbol{s}}_l(k) + \boldsymbol{w}(k) && (76)\\
&= \boldsymbol{U}\widetilde{\boldsymbol{s}}(k) + \boldsymbol{w}(k), && (77)\\
\widetilde{\boldsymbol{s}}_l(k) &= [\,\widetilde{s}_1(k-l)\,\widetilde{s}_2(k-l)\,\cdots\,\widetilde{s}_P(k-l)\,]^T \in \mathbb{C}^P,\\
\boldsymbol{U} &= [\,\boldsymbol{U}_0\,\boldsymbol{U}_1\,\cdots\,\boldsymbol{U}_{L-1}\,]\\
\widetilde{\boldsymbol{s}}(k) &= \left[\,\widetilde{\boldsymbol{s}}_0(k)^T\,\widetilde{\boldsymbol{s}}_1(k)^T\,\cdots\,\widetilde{\boldsymbol{s}}_{L-1}(k)^T\,\right]^T,
\end{aligned}$$

where we also defined the vector $\widetilde{\boldsymbol{s}}(k)$ that collects all source delayed replicas for symbol k. Notice that $\boldsymbol{U}_l$ is an $M \times P$ for $l = 0, 1, \ldots, L-1$.

The data model in eq. (76) describes the whitened samples $\boldsymbol{y}(k)$ as a linear superposition of L echos of the transmitted symbols $\{s_1(k), s_2(k), \ldots, s_P(k)\}$, i.e., the replicas $\widetilde{\boldsymbol{s}}_l(k)$ with delays $l = 0, 1, \ldots, L-1$.

The data $\boldsymbol{y}(k)$ is contaminated by intersymbol interference (ISI) as soon as $L > 1$ (convolutive mixtures). The presence of ISI increases the computational burden of the blind source separation (BSS) algorithms: the echos act as additional virtual sources.

We discuss now in this paragraph a technique that rejects the ISI in the observed samples, i.e., removes the contribution of the echos. More specifically, we outline a procedure that extracts from $\boldsymbol{y}(k)$, L ISI-free signals

$$\boldsymbol{y}_l(k) = \widetilde{\boldsymbol{U}}_l \widetilde{\boldsymbol{s}}_l(k) + \boldsymbol{w}_l(k), \quad l = 0, 1, \ldots, L-1,$$

by exploiting statistical and algebraic properties of the data model, namely that $\widetilde{s}_p(k)$ is white and $\boldsymbol{U}$ is unitary.

Let

$$\boldsymbol{J} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & \ddots & \ddots & 0 \\ 0 & 1 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix}$$

denote the Jordan nilpotent matrix of size $L \times L$. Then, given assumption (A2), we have

$$\boldsymbol{R}_{\widetilde{\boldsymbol{s}}}(l) = \mathrm{E}\left\{\widetilde{\boldsymbol{s}}(k)\widetilde{\boldsymbol{s}}(k-l)^H\right\} = \boldsymbol{J}^l \otimes \boldsymbol{I}_P,$$

for $l > 0$; here, $\otimes$ denotes the Kronecker product. Using this expression in eq. (77), and by assumption (A3),

$$\boldsymbol{R}_{\boldsymbol{y}}(l) = \boldsymbol{U}\boldsymbol{R}_{\widetilde{\boldsymbol{s}}}(l)\boldsymbol{U}^H = \boldsymbol{U}_l\boldsymbol{U}_0^H + \boldsymbol{U}_{l+1}\boldsymbol{U}_1^H + \cdots + \boldsymbol{U}_{L-1}\boldsymbol{U}_{L-1-l}^H, \qquad (78)$$

for $0 < l \leq L-1$. Thus, from (78),

$$\text{span}\{\boldsymbol{R_y}(l)\} = \text{span}\left\{ \left[\begin{array}{cccc} \boldsymbol{U}_l & \boldsymbol{U}_{l+1} & \cdots & \boldsymbol{U}_{L-1} \end{array} \right] \right\}.$$

This means that, through the correlation matrices $\boldsymbol{R_y}(1), \ldots, \boldsymbol{R_y}(L-1)$, the receiver has access to the orthogonal projectors $\boldsymbol{P}_l$ onto span $\{[\boldsymbol{U}_l \cdots \boldsymbol{U}_{L-1}]\}$, for $l = 1, \ldots, L-1$; the lth projector $\boldsymbol{P}_l$ is obtained from the EVD of $\boldsymbol{R_y}(l)$. We can write

$$\boldsymbol{P}_l = \sum_{i=l}^{L-1} \boldsymbol{\Pi}_i,$$

where $\boldsymbol{\Pi}_i = \boldsymbol{U}_i \boldsymbol{U}_i^H$ is the orthogonal projector onto span $\{\boldsymbol{U}_i\}$. In fact, the projectors $\boldsymbol{\Pi}_i$ are the most interesting ones since, due to the orthogonality of $\boldsymbol{U}$,

$$\boldsymbol{\Pi}_l \boldsymbol{U}_m = \boldsymbol{U}_m \delta(l-m).$$

This isolates each replica $\widetilde{\boldsymbol{s}}_l(k)$ from the observations $\boldsymbol{y}(k)$ in eq. (76) as follows:

$$\boldsymbol{y}_l(k) \equiv \boldsymbol{\Pi}_l \boldsymbol{y}(k) = \sum_{m=0}^{L-1} (\boldsymbol{\Pi}_l \boldsymbol{U}_m \widetilde{\boldsymbol{s}}_m(k)) + \boldsymbol{\Pi}_l \boldsymbol{w}(k) = \boldsymbol{U}_l \widetilde{\boldsymbol{s}}_l(k) + \boldsymbol{w}_l(k),$$

where $\boldsymbol{w}_l(k) = \boldsymbol{\Pi}_l \boldsymbol{w}(k)$ denotes the component of $\boldsymbol{w}(k)$ in the subspace span $\{\boldsymbol{U}_l\}$. The projectors $\{\boldsymbol{\Pi}_1, \ldots, \boldsymbol{\Pi}_{L-1}\}$ can be obtained from $\{\boldsymbol{P}_1, \ldots, \boldsymbol{P}_{L-1}\}$ through the identity

$$\boldsymbol{\Pi}_l = \boldsymbol{P}_l \left(\boldsymbol{I}_M - \boldsymbol{P}_{l+1} \right).$$

Also,

$$\boldsymbol{\Pi}_0 = \boldsymbol{I}_M - \boldsymbol{P}_1.$$

3.2 CLOSED FORM SOLUTIONS

We discuss second-order statistics techniques that identify the unknown Multiple-Input Multiple-Output (MIMO) channel analytically, i.e., non-iteratively. These closed-form solutions rely on preprocessing at the transmitter, which judiciously shapes the statistics of the emitted signals.

Here, although we restrict ourselves to multiuser SOS techniques, we should notice that there are available other classes of analytical solutions for the blind channel estimation problem. For single-user scenarios, the work by Tong et al. [37] was the first to accomplish closed-form blind identification of the communication channels based only on SOS. Another SOS based technique is the so-called subspace method introduced by Moulines et al. [26]. Non-SOS techniques relying on redundant precoders, but with the very important property of being robust with respect to the channel spectral nulls, were developed in [9, 29, 30]. See also [24], for precoding techniques in the context of undersampled multiuser communication systems.

3.2.1 Outer-product method

The outer-product decomposition algorithm (OPDA) was introduced in [7]. It identifies the unknown coefficients of the channel matrix up to a residual unitary $P \times P$ ambiguity matrix. This residual instantaneous mixing matrix is solved in closed-form by assuming prior knowledge of certain parts of the composite signal channel, e.g., the pulse shaping filters of the users [8]. Consider the signal model in eq. (65), and let, for the sake of clarity, $L_1 = \cdots = L_P = L$, i.e., all users' channels have the same memory. We start once again with the model in eq. (65) herein repeated for the sake of completeness

$$\boldsymbol{x}(k) = \sum_{p=1}^{P} \boldsymbol{H}_p \boldsymbol{s}_p(k) + \boldsymbol{n}(k), \tag{79}$$

and written in block notation the N-dimensional vector $\boldsymbol{x}(k)$

$$\boldsymbol{x}(k) = \boldsymbol{H}\boldsymbol{s}(k) + \boldsymbol{n}(k), \tag{80}$$

with the $N \times M$, $M = PL$, channel matrix and the PL-dimensional sources signal vector

$$\begin{aligned} \boldsymbol{H} &= [\,\boldsymbol{H}_1\, \boldsymbol{H}_2 \cdots \boldsymbol{H}_P\,], \\ \boldsymbol{s}(k) &= \left[\,\boldsymbol{s}_1(k)^T\, \boldsymbol{s}_2(k)^T \cdots \boldsymbol{s}_P(k)^T\,\right]^T. \end{aligned}$$

The vectors $\boldsymbol{s}_p(k)$ are L-dimensional. We rearrange the data with the multipath permutation $\boldsymbol{L}_L^{PL}$ introduced in eq. (50) to get

$$\begin{aligned} \boldsymbol{x}(k) &= \sum_{l=0}^{L-1} \widetilde{\boldsymbol{H}}_l \widetilde{\boldsymbol{s}}_l(k) + \boldsymbol{n}(k), & (81) \\ &= \widetilde{\boldsymbol{H}}\widetilde{\boldsymbol{s}}(k) + \boldsymbol{n}(k) & (82) \end{aligned}$$

where

$$\begin{aligned} \widetilde{\boldsymbol{H}} &= \left[\widetilde{\boldsymbol{H}}_0\, \widetilde{\boldsymbol{H}}_1 \cdots \widetilde{\boldsymbol{H}}_{L-1}\right] = \boldsymbol{H}\boldsymbol{L}_P^{PL} & (83) \\ \widetilde{\boldsymbol{H}}_l &= \left[\widetilde{\boldsymbol{h}}_1(l)\, \widetilde{\boldsymbol{h}}_2(l) \cdots \widetilde{\boldsymbol{h}}_P(l)\right] & (84) \\ \widetilde{\boldsymbol{s}}(k) &= \boldsymbol{L}_L^{PL}\boldsymbol{s}(k) = \left[\widetilde{\boldsymbol{s}}_0(k)^T \cdots \widetilde{\boldsymbol{s}}_{L-1}(k)^T\right]^T & (85) \\ \widetilde{\boldsymbol{s}}_l(k) &= [s_1(k-l)\, s_2(k-l) \cdots s_P(k-l)\,]^T & (86) \end{aligned}$$

The dimensions of these objects are: $\widetilde{\boldsymbol{H}} : N \times PL$; $\widetilde{\boldsymbol{H}}_l : N \times P$; $\widetilde{\boldsymbol{s}}(k) : PL \times 1$; and $\widetilde{\boldsymbol{s}}_l(k) : P \times 1$. We now stack J successive observations in the NJ-dimensional vector $\boldsymbol{x}_J(k)$

$$\boldsymbol{x}_J(k) = \left[\begin{array}{cccc} \boldsymbol{x}(k)^T & \boldsymbol{x}(k-1)^T & \cdots & \boldsymbol{x}(k-J+1)^T \end{array}\right]^T, \tag{87}$$

Likewise, we consider the $P(L+J-1)$-dimensional signal vector $\widetilde{\boldsymbol{s}}_{L+J-1}(k)$, the NJ-dimensional noise vector $\boldsymbol{n}_J(k)$, and the $NJ \times P(L+J-1)$ block Toeplitz channel matrix $\widetilde{\mathbf{H}}$

$$\begin{aligned}
\widetilde{\boldsymbol{s}}_{L+J-1}(k) &= \left[\begin{array}{cccc} \widetilde{\boldsymbol{s}}(k)^T & \widetilde{\boldsymbol{s}}(k-1)^T & \cdots & \widetilde{\boldsymbol{s}}(k-(L+J-2))^T \end{array} \right]^T \\
\boldsymbol{n}_J(k) &= \left[\begin{array}{cccc} \boldsymbol{n}(k)^T & \boldsymbol{n}(k-1)^T & \cdots & \boldsymbol{n}(k-J+1)^T \end{array} \right]^T \\
\widetilde{\mathbf{H}} &= \left[\begin{array}{ccccccc} \widetilde{\boldsymbol{H}}_0 & \widetilde{\boldsymbol{H}}_1 & \cdots & \widetilde{\boldsymbol{H}}_{L-1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \widetilde{\boldsymbol{H}}_0 & \widetilde{\boldsymbol{H}}_1 & \cdots & \widetilde{\boldsymbol{H}}_{L-1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \widetilde{\boldsymbol{H}}_0 & \widetilde{\boldsymbol{H}}_1 & \cdots & \widetilde{\boldsymbol{H}}_{L-1} \end{array} \right]
\end{aligned} \tag{88}$$

We have

$$\boldsymbol{x}_J(k) = \widetilde{\mathbf{H}} \widetilde{\boldsymbol{s}}_{L+J-1}(k) + \boldsymbol{n}_J(k). \tag{89}$$

The block Toeplitz matrix $\widetilde{\mathbf{H}}$ is parameterized by the $NL \times P$ block generator matrix

$$\boldsymbol{\mathcal{H}} = \left[\begin{array}{c} \widetilde{\boldsymbol{H}}_0 \\ \widetilde{\boldsymbol{H}}_1 \\ \vdots \\ \widetilde{\boldsymbol{H}}_{L-1} \end{array} \right], \tag{90}$$

which collects all the unknown channel blocks.

The OPDA estimates $\boldsymbol{\mathcal{H}}$ based on the following observation. Let the $NL \times (L+J-1)P$ dimensional matrix $\widetilde{\mathbf{H}}_0$ be block Hankel given by

$$\widetilde{\mathbf{H}}_0 = \left[\begin{array}{ccccccccc} \widetilde{\boldsymbol{H}}_0 & \widetilde{\boldsymbol{H}}_1 & \widetilde{\boldsymbol{H}}_2 & \cdots & \widetilde{\boldsymbol{H}}_{L-1} & \mathbf{0} & \cdots & \mathbf{0} \\ \widetilde{\boldsymbol{H}}_1 & \widetilde{\boldsymbol{H}}_2 & \cdots & \widetilde{\boldsymbol{H}}_{L-1} & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ \widetilde{\boldsymbol{H}}_2 & \cdots & \widetilde{\boldsymbol{H}}_{L-1} & \mathbf{0} & \cdots & & \cdots & \mathbf{0} \\ \vdots & & & & & & & \vdots \\ \widetilde{\boldsymbol{H}}_{L-1} & \mathbf{0} & \cdots & \cdots & \cdots & \cdots & \cdots & \mathbf{0} \end{array} \right]. \tag{91}$$

If follows that the Hermitian matrix

$$\boldsymbol{D}_1 = \widetilde{\mathbf{H}}_0 \widetilde{\mathbf{H}}_0^H \tag{92}$$

can be written as

$$\boldsymbol{D}_1 = \left[\begin{array}{cccc} \boldsymbol{D}_{11} & \boldsymbol{D}_{12} & \cdots & \boldsymbol{D}_{1L} \\ \boldsymbol{D}_{21} & \boldsymbol{D}_{22} & \cdots & \boldsymbol{D}_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{D}_{L1} & \boldsymbol{D}_{L2} & \cdots & \boldsymbol{D}_{LL} \end{array} \right] \qquad \text{with} \qquad \boldsymbol{D}_{ij} = \sum_{k=i-1}^{L-1} \widetilde{\boldsymbol{H}}_k \widetilde{\boldsymbol{H}}_{k+j-1}^H,$$

with $\widetilde{\boldsymbol{H}}_k = \mathbf{0}$, for $k > L-1$. Defining from the lower right block of $\boldsymbol{D}_1$

$$\boldsymbol{D}_2 = \begin{bmatrix} \boldsymbol{D}_{22} & \cdots & \boldsymbol{D}_{1L} & \mathbf{0} \\ \boldsymbol{D}_{32} & \cdots & \boldsymbol{D}_{2L} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{D}_{L1} & \cdots & \boldsymbol{D}_{LL} & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{bmatrix},$$

we have the important result:

$$\boldsymbol{\Delta} D = \boldsymbol{D}_1 - \boldsymbol{D}_2 = \boldsymbol{\mathcal{H}}\boldsymbol{\mathcal{H}}^H.$$

Thus, an SVD of the outer-product matrix $\boldsymbol{\Delta} D$ yields $\boldsymbol{\mathcal{H}}$ up to a unitary matrix, i.e.,

$$\widehat{\boldsymbol{\mathcal{H}}} = \boldsymbol{\mathcal{H}}\boldsymbol{Q}^H,$$

where the $P \times P$ matrix $\boldsymbol{Q}$ is unitary.

To implement this approach we need to estimate only the Hermitian matrix $\boldsymbol{D}_1$. We show next that this matrix is obtained directly from the correlation matrices of the system outputs, i.e., from

$$\left\{ \boldsymbol{R}_{\boldsymbol{x}}(l) = \mathrm{E}\left\{ \boldsymbol{x}(k)\boldsymbol{x}(k-l)^H \right\},\ l = 0, 1, \cdots, L-1 \right\}.$$

These correlations can be computed to be

$$\boldsymbol{R}_{\boldsymbol{x}}(l) = \sum_{i=0}^{L-1} \widetilde{\boldsymbol{H}}_{i+l} \widetilde{\boldsymbol{H}}_i^H. \tag{93}$$

Notice also that

$$\mathbf{R}_{\boldsymbol{x}_J} = \mathrm{E}\left\{ \boldsymbol{x}_J(k)\boldsymbol{x}_J(k)^H \right\} = \begin{bmatrix} \boldsymbol{R}_{\boldsymbol{x}}(0) & \boldsymbol{R}_{\boldsymbol{x}}(1) & \cdots & \boldsymbol{R}_{\boldsymbol{x}}(J-1) \\ \boldsymbol{R}_{\boldsymbol{x}}(1)^H & \boldsymbol{R}_{\boldsymbol{x}}(0) & \cdots & \boldsymbol{R}_{\boldsymbol{x}}(J-2) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{R}_{\boldsymbol{x}}(J-1)^H & \boldsymbol{R}_{\boldsymbol{x}}(J-2)^H & \cdots & \boldsymbol{R}_{\boldsymbol{x}}(0) \end{bmatrix} \tag{94}$$

$$= \widetilde{\mathbf{H}}\widetilde{\mathbf{H}}^H + \sigma^2 \boldsymbol{I}_{NJ}. \tag{95}$$

The last equation follows by direct computation from eq. (87).Further, define the following block Hankel matrix

$$\mathbf{R}_0 = \begin{bmatrix} \boldsymbol{R}_{\boldsymbol{x}}(0) - \sigma^2 \boldsymbol{I}_N & \boldsymbol{R}_{\boldsymbol{x}}(1) & \cdots & \boldsymbol{R}_{\boldsymbol{x}}(L-1) \\ \boldsymbol{R}_{\boldsymbol{x}}(1) & \boldsymbol{R}_{\boldsymbol{x}}(2) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{R}_{\boldsymbol{x}}(L-1) & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}.$$

Replacing the entries of matrix $\mathbf{R}_0$ by their definition from eq. (93) for $l = 0, \cdots, L-1$, and, after some algebra, we can see that

$$\mathbf{R}_0 = \widetilde{\mathbf{H}}_0 \widetilde{\mathbf{H}}^H, \tag{96}$$

where $\widetilde{\mathbf{H}}$ and $\widetilde{\mathbf{H}}_0$ are defined in eq. (88) and eq. (91), respectively.

Since from the assumptions on the channel we can assume that $\widetilde{\mathbf{H}}$ is full column rank, its pseudoinverse $\widetilde{\mathbf{H}}^\dagger = \widetilde{\mathbf{H}}^H \left(\widetilde{\mathbf{H}}\widetilde{\mathbf{H}}^H\right)^\dagger$ is well defined and $\widetilde{\mathbf{H}}^\dagger\widetilde{\mathbf{H}} = \boldsymbol{I}$. We can then go back to the definition of $\boldsymbol{D}_1$ from eq.(92), and obtain successively

$$\begin{aligned}
\boldsymbol{D}_1 &= \mathbf{H}_0\widetilde{\mathbf{H}}_0^H && (97)\\
&= \widetilde{\mathbf{H}}_0 \left(\widetilde{\mathbf{H}}^\dagger\widetilde{\mathbf{H}}\right)\widetilde{\mathbf{H}}_0^H && (98)\\
&= \widetilde{\mathbf{H}}_0\widetilde{\mathbf{H}}^H \left(\widetilde{\mathbf{H}}\widetilde{\mathbf{H}}^H\right)^\dagger \widetilde{\mathbf{H}}\widetilde{\mathbf{H}}_0^H && (99)\\
&= \mathbf{R}_0 \left(\mathbf{R}_{\boldsymbol{x}_J} - \sigma^2\boldsymbol{I}_{NJ}\right)^\dagger \mathbf{R}_0^H. && (100)
\end{aligned}$$

Eq.(98) follows by inserting the identity matrix in between the two factors of eq.(97) and replacing it by the product $\widetilde{\mathbf{H}}^\dagger\widetilde{\mathbf{H}}$; eq.(99) follows by replacing the pseudoinverse by its definition; eq.(100) recognizes from eq.(96) the first two factors and the last two factors as $\mathbf{R}_0$ and its Hermitian, and recognizes from eq.(95) the middle factor as $\left(\mathbf{R}_{\boldsymbol{x}_J} - \sigma^2\boldsymbol{I}_{NJ}\right)^\dagger$.

Thus, in practice, $\boldsymbol{D}_1$ is estimated via (100), with the correlation matrices $\boldsymbol{R}_{\boldsymbol{x}}(l)$ needed in $\mathbf{R}_{\boldsymbol{x}_J}$ and $\mathbf{R}_0$, replaced by their consistent estimators

$$\widehat{\boldsymbol{R}}_{\boldsymbol{x}}(l) = \frac{1}{K-l}\sum_{k=l+1}^{K} \boldsymbol{x}(k)\boldsymbol{x}(k-l)^H,$$

where K denotes the number of samples $\boldsymbol{x}(k)$ available.

We note that other blind SOS-techniques can also convert a convolutive mixture of P users into an instantaneous one, i.e., the matrix $\boldsymbol{H}$ is solved up to a $P \times P$ unknown mixing matrix. We refer the interested reader to the methods in [39, 40, 41] that achieve this goal even for infinite-impulse response (IIR) channels with minimum-phase common zeros among the subchannels. See also [15, 25] where the MIMO channel can be recovered up to a block diagonal constant matrix; in the special case where the users are exposed to distinct channel degrees, then the convolutive mixture is completely resolved.

It remains solving for the residual instantaneous mixture matrix $\boldsymbol{Q}$. As described in [8], this can be accomplished analytically if part of the continuous-time composite channel $h_p(t)$ for the pth user is known a priori by the receiver.

Recall the derivation of the data model in section 1.1, and assume, for simplicity, $N_a = 1 \Rightarrow N = J$. The composite channel $h_p(t)$ is given by the convolution of a pulse-shaping filter $u_p(t)$ with a propagation channel $c_p(t)$:

$$h_p(t) = u_p \star c_p(t), \tag{101}$$

where $\star$ denotes convolution and $c_p(t) \equiv h_{\mathrm{cha}_{pn}} \star h_{\mathrm{rec}}(t)$ (for $n = 1$), $h_{\mathrm{rec}}(t)$ being the receiver filter impulse response. In many communication systems, knowledge of $u_p(t)$ is available at the receiver; in this case, only the unknown part $c_p(t)$ needs to be estimated.

Assume the data is acquired with an oversampling factor of J, i.e., the sampling period is $T_s = T/J$, where T is the symbol baud period. The $J \times 1$ vectors $\widetilde{\boldsymbol{h}}_p(l)$ in (84) are given by

$$\widetilde{\boldsymbol{h}}_p(l) = [h_p((l-1)T) \;\; h_p((l-1)T + T_s) \;\; \cdots \;\; h_p((l-1)T + (J-1)T_s)]^T,$$

i.e., $\widetilde{\boldsymbol{h}}_p(l)$ takes J equi-spaced samples of the continuous time channel $h_p(t)$ from the slot $[(l-1)T; lT)$. We now relate $\widetilde{\boldsymbol{h}}_p(l)$ with the $JL \times P$ block matrix $\boldsymbol{\mathcal{H}}$ given in eq. (90) that collects all the channel unknowns. Reexpressing $\boldsymbol{\mathcal{H}}$ in terms of its columns

$$\boldsymbol{\mathcal{H}} = \left[\mathfrak{h}_1 \cdots \mathfrak{h}_p \cdots \mathfrak{h}_P\right].$$

The pth-column $\mathfrak{h}_p$ of $\boldsymbol{\mathcal{H}}$ is the NL-dimensional multichannel vector for the pth user

$$\mathfrak{h}_p = \left[\begin{array}{cccc} \widetilde{\boldsymbol{h}}_p(0)^T & \widetilde{\boldsymbol{h}}_p(1)^T & \cdots & \widetilde{\boldsymbol{h}}_p(L-1)^T \end{array}\right]^T.$$

In other words, the pth column of $\boldsymbol{\mathcal{H}}$ contains the T_s-sampled response $h_p(t)$ for source p. The discrete counterpart of eq. (101) is the discrete convolution given in matrix format by

$$\mathfrak{h}_p = \boldsymbol{\mathcal{U}}_p \mathfrak{c}_p, \tag{102}$$

where

$$\boldsymbol{\mathcal{U}}_p = \left[\begin{array}{cccc} u_p(0) & 0 & \cdots & 0 \\ u_p(1) & u_p(0) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ u_p(L_f) & \ddots & \ddots & u_p(0) \\ 0 & u_p(L_f) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & u_p(L_f) \end{array}\right],$$

$u_p(i) \equiv u_p(iT_s)$, and

$$\mathfrak{c}_p = \left[\begin{array}{ccccc} \boldsymbol{c}_p(0) & \boldsymbol{c}_p(1) & \boldsymbol{c}_p(2) & \cdots & \boldsymbol{c}_p(L_c) \end{array}\right]^T,$$

$\boldsymbol{c}_p(i) \equiv c_p(iT_s)$, for appropriate integers L_f, L_c. Thus, $\boldsymbol{\mathcal{H}}$ can be written as

$$\boldsymbol{\mathcal{H}} = \left[\begin{array}{cccc} \boldsymbol{\mathcal{U}}_1\mathfrak{c}_1 & \boldsymbol{\mathcal{U}}_2\mathfrak{c}_2 & \cdots & \boldsymbol{\mathcal{U}}_P\mathfrak{c}_P \end{array}\right].$$

Denote the OPDA's estimate of $\boldsymbol{\mathcal{H}}$ by

$$\widehat{\boldsymbol{\mathcal{H}}} = \boldsymbol{\mathcal{H}}\boldsymbol{Q}^H$$

with the $P \times P$ matrix $\boldsymbol{Q} = \left[\boldsymbol{q}_1 \cdots \boldsymbol{q}_p\right]$ unitary. It follows that

$$\widehat{\boldsymbol{\mathcal{H}}}\boldsymbol{q}_p = \boldsymbol{\mathcal{U}}_p\boldsymbol{c}_p,$$

or, equivalently,

$$\underbrace{\left[\boldsymbol{\mathcal{U}}_p - \widehat{\boldsymbol{\mathcal{H}}}\right]}_{\boldsymbol{S}_p} \begin{bmatrix} \boldsymbol{c}_p \\ \boldsymbol{q}_p \end{bmatrix} = \mathbf{0}, \tag{103}$$

i.e., the unknowns $(\boldsymbol{c}_p, \boldsymbol{q}_p)$ belong to the kernel of the matrix $\boldsymbol{S}_p$, which is available at the receiver. If the homogeneous system in (103) has a unique solution, then this yields a simple method for finding the unknowns. As shown in [8], uniqueness of solutions is guaranteed provided that the z^{-1}-polynomials $H_i(z)$ $(i \neq p)$ are linearly independent of $U_p(z)L(z)$ for all $L(z)$, where $H_i(z)$ and $U_p(z)$ denote the z-transforms of the discrete-time signals $h_i(\cdot)$ and $u_p(\cdot)$, respectively.

3.2.2 Transmitter induced conjugate cyclostationarity

The transmitter induced conjugate cyclostationarity (TICC) technique for blind identification of MIMO systems was introduced in [6]. It consists in inducing a cyclic frequency in each user's information sequence prior to transmission. This diversity in the cyclospectra of the emitted signals is then exploited at the receiver, to reduce the original problem of blind MIMO identification to P simpler blind Single-Input Multiple-Output (SIMO) identification ones. To solve the latter problem, standard algorithms, e.g., the subspace method [26], can be employed. Before giving an outline of this technique, we need some definitions.

For an $N \times M$ matrix $\boldsymbol{A} = [\,\boldsymbol{a}_1 \cdots \boldsymbol{a}_M\,]$ and an integer J, we let the $N(J+1) \times (M+J+1)$ matrix $\mathcal{T}_J(\boldsymbol{A})$ denote the block-Sylvester matrix

$$\mathcal{T}_J(\boldsymbol{A}) = \begin{bmatrix} \boldsymbol{a}_1 & \cdots & \boldsymbol{a}_M & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{a}_1 & \cdots & \boldsymbol{a}_M & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{a}_1 & \cdots & \boldsymbol{a}_M \end{bmatrix}. \tag{104}$$

The conjugate cyclocorrelation sequence of a signal $\boldsymbol{z}(k)$ at the frequency α, written $\boldsymbol{R}_{\boldsymbol{z}}^{(\alpha)}(\tau)$, $\tau \in \mathbb{Z}$, is given by

$$\boldsymbol{R}_{\boldsymbol{z}}^{(\alpha)}(\tau) = \lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} \mathrm{E}\left\{\boldsymbol{z}(k+\tau)\boldsymbol{z}(k)^T\right\} e^{-i2\pi\alpha k}. \tag{105}$$

Consider the signal model in (65) and assume, without loss of generality, that $s_p(k)$ denote real i.i.d. symbol sequences with unit power. Cyclic frequencies are induced at the emitter by modulating each information sequence $s_p(k)$ by a complex exponential:

$$\breve{s}_p(k) = s_p(k) e^{i\pi\alpha_p k},$$

with $\alpha_p \neq \alpha_q$ for $p \neq q$, $p, q \in \{1, 2, \ldots, P\}$. Thus, we have

$$\boldsymbol{x}(k) = \sum_{p=1}^{P} \boldsymbol{H}_p \breve{\boldsymbol{s}}_p(k) + \boldsymbol{n}(k),$$

as in (65). Applying definition (105) to the observations $\boldsymbol{x}(k)$ at the cyclic frequency β:

$$\boldsymbol{R}_{\boldsymbol{x}}^{(\beta)}(\tau) = \sum_{p=1}^{P} \boldsymbol{H}_p \boldsymbol{R}_{\boldsymbol{s}_p}(\tau) \breve{\boldsymbol{H}}_p^T \delta(\alpha_p - \beta), \tag{106}$$

where $\breve{\boldsymbol{H}}_p = \left[\breve{\boldsymbol{h}}_p(0)\, \breve{\boldsymbol{h}}_p(1)\, \cdots\, \breve{\boldsymbol{h}}_p(L_p-1)\right]$, $\breve{\boldsymbol{h}}_p(l) = \boldsymbol{h}_p(l) e^{-j2\pi\alpha_p l}$; here, $\delta(\lambda) = 1$ if $\lambda = 0$ and $\delta(\lambda) = 0$ if $\lambda \neq 0$. Equation (106) discloses the important fact that, at the frequency $\beta = \alpha_p$, only the pth user contributes to $\boldsymbol{R}_{\boldsymbol{x}}^{(\alpha_p)}(\tau)$,

$$\boldsymbol{R}_{\boldsymbol{x}}^{(\alpha_p)}(\tau) = \boldsymbol{H}_p \boldsymbol{R}_{\boldsymbol{s}_p}(\tau) \breve{\boldsymbol{H}}_p^T. \tag{107}$$

Let $\boldsymbol{\mathcal{R}}_J^{(\beta)}$ denote the $N(J+1) \times N(J+1)$ block Toeplitz matrix

$$\boldsymbol{\mathcal{R}}_J^{(\beta)} = \begin{bmatrix} \boldsymbol{R}_{\boldsymbol{x}}^{(\beta)}(0) & \boldsymbol{R}_{\boldsymbol{x}}^{(\beta)}(1) & \cdots & \boldsymbol{R}_{\boldsymbol{x}}^{(\beta)}(J) \\ \boldsymbol{R}_{\boldsymbol{x}}^{(\beta)}(-1) & \boldsymbol{R}_{\boldsymbol{x}}^{(\beta)}(0) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \boldsymbol{R}_{\boldsymbol{x}}^{(\beta)}(1) \\ \boldsymbol{R}_{\boldsymbol{x}}^{(\beta)}(-J) & \cdots & \boldsymbol{R}_{\boldsymbol{x}}^{(\beta)}(-1) & \boldsymbol{R}_{\boldsymbol{x}}^{(\beta)}(0) \end{bmatrix}. \tag{108}$$

From (107), we have the equality

$$\boldsymbol{\mathcal{R}}_J^{(\alpha_p)} = \boldsymbol{\mathcal{T}}_J(\boldsymbol{H}_p)\, \boldsymbol{\mathcal{T}}_J\left(\breve{\boldsymbol{H}}_p\right)^T.$$

If $J > L_p$ and $\boldsymbol{h}_p(z) = \sum_{l=0}^{L_p-1} \boldsymbol{h}_p(l) z^{-l} \neq 0$ for all z, then it can been proved that both $\boldsymbol{\mathcal{T}}_J(\boldsymbol{H}_p)$ and $\boldsymbol{\mathcal{T}}_J(\breve{\boldsymbol{H}}_p)$ are full column rank. This implies that the left kernel of $\boldsymbol{\mathcal{R}}_J^{(\alpha_p)}$, i.e.,

$$\mathcal{N}_l\left(\boldsymbol{\mathcal{R}}_J^{(\alpha_p)}\right) = \left\{\boldsymbol{x} \,:\, \boldsymbol{x}^H \boldsymbol{\mathcal{R}}_J^{(\alpha_p)} = \boldsymbol{0}\right\},$$

coincides with the left kernel of $\boldsymbol{\mathcal{T}}_J(\boldsymbol{H}_p)$. Let $\boldsymbol{\Pi}$ denote the orthogonal projector onto $\mathcal{N}_l\left(\boldsymbol{\mathcal{R}}_J^{(\alpha_p)}\right)$; this can be obtained from its SVD. It follows that the linear homogeneous system

$$\boldsymbol{\Pi}\boldsymbol{\mathcal{T}}_J(\boldsymbol{X}) = \boldsymbol{0} \tag{109}$$

in the unknown $\boldsymbol{X} = \left[\boldsymbol{x}_0\, \boldsymbol{x}_1\, \cdots\, \boldsymbol{x}_{L_p-1}\right]$ yields the solution

$$\boldsymbol{X} = \lambda \boldsymbol{H}_p,$$

for some scalar λ [26]. The scalar λ can also be resolved, see [6] for details. Thus, in practice, given a finite set of K data samples, the TICC technique solves P linear systems (109) for $p = 1, \ldots, P$; the matrix $\boldsymbol{\mathcal{R}}_J^{(\beta)}$ in (108) is estimated by replacing $\boldsymbol{R}_{\boldsymbol{x}}^{(\beta)}(\tau)$ with the consistent estimate

$$\widehat{\boldsymbol{R}}_{\boldsymbol{x}}^{(\beta)}(\tau) = \frac{1}{K} \sum_{k=1}^{K-\tau} \boldsymbol{x}(k+\tau) \boldsymbol{x}(k)^T e^{-j2\pi\beta k}.$$

3.2.3 CFC$_2$: Closed-form correlative coding

The closed-form correlative coding (CFC$_2$) method for MIMO blind channel identification was formulated in [47]. The main idea consists in pre-filtering the data symbols before transmission, in order to induce a certain diversity in their SOS. This inserted structure enables the receiver to recover the MIMO channel by a closed form solution. The need for coloring the emitted information sequences is a consequence of a well-known fact: in general, the MIMO channel matrix $\boldsymbol{H}$ (recall the data model in (65)) cannot be unambiguously determined from the SOS of the MIMO output, i.e., from the set of correlation matrices

$$\mathcal{R}_{\boldsymbol{x}} = \{\boldsymbol{R}_{\boldsymbol{x}}(\tau) \,:\, \tau \in \mathbb{Z}\}.$$

To make explicit the channel matrix $\boldsymbol{H}$, we will also refer to this set as $\mathcal{R}_{\boldsymbol{x}}(\boldsymbol{H})$.

For an illustrative example, consider the $4 \times (2+2)$ case $\boldsymbol{H} = [\,\boldsymbol{H}_1\, \boldsymbol{H}_2\,]$. Then, it is easily seen that, under Assumption (A2) that the $s_p(k)$ are mutually independent uncorrelated symbols, the following holds:

$$\mathcal{R}_{\boldsymbol{x}}(\boldsymbol{H}) = \mathcal{R}_{\boldsymbol{x}}(\boldsymbol{H}\boldsymbol{Q}), \tag{110}$$

where the 4×4 matrix $\boldsymbol{Q}$ is a unitary mixing matrix given by

$$\boldsymbol{Q} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \otimes \boldsymbol{I}_2.$$

Equation (110) says that the mapping

$$\boldsymbol{H} \mapsto \mathcal{R}_{\boldsymbol{x}}(\boldsymbol{H})$$

is not one-to-one in a non-trivial way: both the "true" MIMO channel matrix $\boldsymbol{H}$ and a mixed version of it

$$\widetilde{\boldsymbol{H}} \equiv \boldsymbol{H}\boldsymbol{Q}$$

induce the same SOS at the MIMO output.

We show that $\widetilde{\boldsymbol{H}}$ is useless for source separation as the two users are still mixed. To see this, and discarding the noise, we have working with $\boldsymbol{H}$

$$\widehat{\boldsymbol{s}}(k) = \boldsymbol{H}^{\dagger}\boldsymbol{x}(k) = \boldsymbol{s}(k) = \begin{bmatrix} s_1(k) \\ s_1(k-1) \\ s_2(k) \\ s_2(k-1) \end{bmatrix},$$

i.e., the sources symbols are separated, whereas, working with $\widetilde{\boldsymbol{H}}$

$$\widehat{\boldsymbol{s}}(k) = \widetilde{\boldsymbol{H}}^{\dagger}\boldsymbol{x}(k) = \frac{1}{\sqrt{2}} \begin{bmatrix} s_1(k) + s_2(k) \\ s_1(k-1) + s_2(k-1) \\ -s_1(k) + s_2(k) \\ -s_1(k-1) + s_2(k-1) \end{bmatrix},$$

and the data symbols are still mixed.

In summary, the problem of identification of $\boldsymbol{H}$ from $\mathcal{R}_{\boldsymbol{x}}$ is not a well-posed problem, under the standard assumptions (A1)-(A3). Based on this observation, the CFC$_2$ technique replaces the original (white) symbol sequences, renamed now to $a_p(k)$, by a filtered (colored) version $s_p(k)$ of it:

$$s_p(k) = \sum_{m=0}^{M_p-1} c_p(m) a_p(k-m),$$

where $\{c_p(m) : m = 0, 1, \ldots, M_p - 1\}$ is the finite-impulsive response (FIR) of the pth correlative filter. In order to maintain the transmitted power, we restrict ourselves to unit-power correlative filters, i.e.,

$$\forall p = 1, 2, \ldots, P: \qquad \sum_{m=0}^{M_p-1} |c_p(m)|^2 = 1.$$

The pre-filters $c_p(\cdot)$ change the SOS of $s_p(k)$, and, as a consequence, of the vector process

$$\boldsymbol{s}_p(k) = [\, s_p(k)\, s_p(k-1)\, \cdots\, s_p(k-L_p+1)\,]^T.$$

Denote by

$$\boldsymbol{\Gamma}_p(\tau) = \boldsymbol{R}_{\boldsymbol{s}_p}(0)^{-1/2} \boldsymbol{R}_{\boldsymbol{s}_p}(\tau) \boldsymbol{R}_{\boldsymbol{s}_p}(0)^{-1/2}, \tag{111}$$

the (normalized) correlation matrix of $\boldsymbol{s}_p(k)$ at lag τ; this generalizes the correlation coefficient of a scalar wide-sense stationary (WSS) random process $x(k)$:

$$\gamma_x(\tau) = \frac{\mathrm{E}\,\{x(k)x(k-\tau)^*\}}{\sqrt{\mathrm{var}\{x(k)\}}\sqrt{\mathrm{var}\{x(k-\tau)\}}} = r_x(0)^{-1/2} r_x(\tau) r_x(0)^{-1/2}.$$

Suppose that the filters $c_p(\cdot)$ are designed such that the following condition holds:

(A4) for all $p \neq q$, there is a correlation lag $\tau = \tau(p,q)$ such that

$$\sigma\,\{\boldsymbol{\Gamma}_p(\tau)\} \cap \sigma\,\{\boldsymbol{\Gamma}_q(\tau)\} = \emptyset, \tag{112}$$

where $\sigma\,\{\boldsymbol{A}\} \subset \mathbb{C}$ denotes the set of eigenvalues of $\boldsymbol{A}$ (the spectrum of $\boldsymbol{A}$).

Assumption (A4) requires that, for two distinct sources p and q, condition (112) must hold for *some* $\tau(p,q)$; it is not necessary, that it holds for *all* $\tau \in \mathbb{Z}$. Moreover, if (p,q) changes, $\tau(p,q)$ is allowed to change. This makes the set of correlative filters that satisfy (A4) very generic, irrespective of the length $M_p > 1$, even if we further restrict ourselves to minimum-phase filters, we refer the interested reader to [50]; the minimum-phase property is attractive for sub-optimal direct inversion of $c_p(\cdot)$ when recovering the input information sequence $a_p(k)$, after $\boldsymbol{H}$ has been identified. The CFC$_2$ method relies on the following identifiability result.

Theorem 3.2 [Identifiability] *Consider that the signal model in eq. (65) and assumptions (A1)-(A4) hold. Then, the mapping $\boldsymbol{H} \mapsto \mathcal{R}_{\boldsymbol{x}}(\boldsymbol{H})$ is one-to-one up to a phase offset per user; i.e., if $\mathcal{R}_{\boldsymbol{x}}(\boldsymbol{H}) = \mathcal{R}_{\boldsymbol{x}}(\widetilde{\boldsymbol{H}})$, then $\widetilde{\boldsymbol{H}}_p = \boldsymbol{H}_p e^{j\theta_p}$, for some $\theta_p \in \mathbb{R}$, $p = 1, 2, \ldots, P$.* □

The proof is found in [50].

Notice that the phase ambiguity per user cannot be avoided as only SOS are considered.

In the following, we focus on the closed-form algorithm that identifies $\boldsymbol{H}$ from the set of correlation matrices $\mathcal{R}_{\boldsymbol{x}}$. The goal is to exploit the *Identifiability Theorem* 2.2 and find a matrix $\widetilde{\boldsymbol{H}}$ such that $\mathcal{R}_{\boldsymbol{x}} = \mathcal{R}_{\boldsymbol{x}}(\widetilde{\boldsymbol{H}})$, i.e.,

$$\boldsymbol{R}_{\boldsymbol{x}}(\tau) = \widetilde{\boldsymbol{H}}\boldsymbol{R}_{\boldsymbol{s}}(\tau)\widetilde{\boldsymbol{H}}^H + \boldsymbol{R}_{\boldsymbol{n}}(\tau), \tag{113}$$

for all $\tau \in \mathbb{Z}$. If this is achieved, then the *Identifiability Theorem* 3.2 guarantees that $\widetilde{\boldsymbol{H}} = \boldsymbol{H}$ up to a phase offset per user. Let

$$\boldsymbol{R}(\tau) = \boldsymbol{R}_{\boldsymbol{x}}(\tau) - \boldsymbol{R}_{\boldsymbol{n}}(\tau)$$

denote the denoised correlation matrices of the MIMO channel output. Since all filters have finite memory, it suffices that (113) holds for a finite set of τ's, say $\mathcal{T} = \{\tau_1, \tau_2, \ldots, \tau_I\}$. Thus, the problem is reduced to finding $\widetilde{\boldsymbol{H}}$ such that the factorization

$$\forall \tau \in \mathcal{T}: \qquad \boldsymbol{R}(\tau) = \widetilde{\boldsymbol{H}}\boldsymbol{R}_{\boldsymbol{s}}(\tau)\widetilde{\boldsymbol{H}}^H,$$

holds. The algorithm solves for $\widetilde{\boldsymbol{H}}$ in three steps.

Step 1. Let $\boldsymbol{R}(0) = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^H$ be an EVD of $\boldsymbol{R}(0)$ where the $N \times N$ matrix $\boldsymbol{V}$ is unitary and

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Lambda}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix},$$

where $\boldsymbol{\Lambda}_1 = \operatorname{diag}(\lambda_1, \ldots, \lambda_M)$, $\lambda_1 \geq \cdots \geq \lambda_M > 0$. Define the $N \times M$ matrix $\boldsymbol{G}_0$ by $\boldsymbol{G}_0 = \boldsymbol{V}_1\boldsymbol{\Lambda}_1^{1/2}$, where the $N \times M$ matrix $\boldsymbol{V}_1$ consists of the first M columns of $\boldsymbol{V}$. We remark that $\boldsymbol{G}_0$, thus defined, satisfies the identity

$$\boldsymbol{G}_0 = \underbrace{\boldsymbol{H}\boldsymbol{R}_{\boldsymbol{s}}(0)^{1/2}}_{\boldsymbol{H}_0}\boldsymbol{Q}^H,$$

where the $M \times M$ unitary matrix $\boldsymbol{Q}$ is unknown. This results from the fact that both $\boldsymbol{G}_0$ and $\boldsymbol{H}_0$ are full column-rank matrices verifying

$$\boldsymbol{R}(0) = \boldsymbol{G}_0\boldsymbol{G}_0^H = \boldsymbol{H}_0\boldsymbol{H}_0^H.$$

Step 2. From this observation, it follows that

$$\boldsymbol{B}(\tau) \equiv \boldsymbol{G}_0^\dagger \boldsymbol{R}(\tau)\boldsymbol{G}_0^{\dagger H}$$

satisfies

$$\boldsymbol{B}(\tau) = \left[\boldsymbol{Q}\boldsymbol{R}_{\boldsymbol{s}}(0)^{-1/2}\boldsymbol{H}^\dagger\right]\left[\boldsymbol{H}\boldsymbol{R}_{\boldsymbol{s}}(\tau)\boldsymbol{H}^H\right]\left[\boldsymbol{H}^{\dagger H}\boldsymbol{R}_{\boldsymbol{s}}(0)^{-1/2}\boldsymbol{Q}^H\right] = \boldsymbol{Q}\boldsymbol{\Gamma}(\tau)\boldsymbol{Q}^H, \tag{114}$$

where

$$\boldsymbol{\Gamma}(\tau) \equiv \boldsymbol{R_s}(0)^{-1/2}\boldsymbol{R_s}(0)\boldsymbol{R_s}(0)^{-1/2}.$$

Since all the $\boldsymbol{R_s}(\tau)$ are block diagonal,

$$\boldsymbol{R_s}(\tau) = \mathrm{diag}\left(\boldsymbol{R}_{\boldsymbol{s}_1}(\tau), \ldots, \boldsymbol{R}_{\boldsymbol{s}_P}(\tau)\right),$$

then $\boldsymbol{\Gamma}(\tau)$ is also block diagonal

$$\boldsymbol{\Gamma}(\tau) = \mathrm{diag}\left(\boldsymbol{\Gamma}_1(\tau), \ldots, \boldsymbol{\Gamma}_P(\tau)\right),$$

where $\boldsymbol{\Gamma}_p(\tau)$ was defined in (111). Thus, letting $\boldsymbol{Q} = [\,\boldsymbol{Q}_1 \cdots \boldsymbol{Q}_P\,]$, equation (114) reads as

$$\boldsymbol{B}(\tau) = \sum_{p=1}^{P} \boldsymbol{Q}_p \boldsymbol{\Gamma}_p(\tau) \boldsymbol{Q}_p^H.$$

Due to the orthogonality of $\boldsymbol{Q}$,

$$\boldsymbol{Q}_p^H \boldsymbol{Q}_q = \boldsymbol{I}_L \delta(p-q),$$

and $\boldsymbol{X} = \boldsymbol{Q}_p$ is a solution of the pth linear system:

$$\left\{ \begin{array}{rcl} \boldsymbol{B}(\tau_1)\boldsymbol{X} - \boldsymbol{X}\boldsymbol{\Gamma}_p(\tau_1) &=& \boldsymbol{0} \\ & \vdots & \\ \boldsymbol{B}(\tau_I)\boldsymbol{X} - \boldsymbol{X}\boldsymbol{\Gamma}_p(\tau_I) &=& \boldsymbol{0} \end{array} \right. \quad \Leftrightarrow \quad \boldsymbol{B}(\tau)\boldsymbol{X} - \boldsymbol{X}\boldsymbol{\Gamma}_p(\tau) = \boldsymbol{0}, \tau \in \mathcal{T}. \tag{115}$$

It turns out that $\boldsymbol{Q}_p$ is the *unique* solution (up to a scalar factor) of eq. (109), see [50].

Since $\boldsymbol{B}(\tau)$ and $\boldsymbol{\Gamma}_p(\tau)$ are available to the receiver, this provides a way to identify $\boldsymbol{Q}_p$. Thus, let $\boldsymbol{X}$ be a non-zero solution of eq. (115) and re-scale as

$$\boldsymbol{U}_p = \boldsymbol{X}/(\|\boldsymbol{X}\|\sqrt{L}).$$

It follows that

$$\boldsymbol{U}_p = \boldsymbol{Q}_p e^{j\theta_p}, \qquad \text{for some} \quad \theta_p \in \mathbb{R}.$$

It is straightforward to solve the homogeneous system (115) in a least-squares sense. Let

$$\begin{aligned} f(\boldsymbol{X}) &= \sum_{\tau \in \mathcal{T}} \|\boldsymbol{B}(\tau)\,\boldsymbol{X} - \boldsymbol{X}\boldsymbol{\Gamma}_p(\tau)\|^2 \\ &= \sum_{i=1}^{I} \left\| \left(\boldsymbol{I}_L \otimes \boldsymbol{B}(\tau_i) - \boldsymbol{\Gamma}_p(\tau_i)^T \otimes \boldsymbol{I}_M \right) \boldsymbol{x} \right\|^2 \\ &= \|\boldsymbol{T}\boldsymbol{x}\|^2 \\ &= \boldsymbol{x}^H \boldsymbol{T}^H \boldsymbol{T} \boldsymbol{x}, \end{aligned} \tag{116}$$

where $\boldsymbol{x} = \mathrm{vec}\,(\boldsymbol{X})$ and

$$\begin{aligned} \boldsymbol{T} &= \left[\boldsymbol{T}_1^T\, \boldsymbol{T}_2^T\, \cdots\, \boldsymbol{T}_k^T\right]^T \\ \boldsymbol{T}_i &= \boldsymbol{I}_L \otimes \boldsymbol{B}\,(\tau_i) - \boldsymbol{\Gamma}_p\,(\tau_i)^T \otimes \boldsymbol{I}_M . \end{aligned}$$

A global minimizer of $f\,(\boldsymbol{X})$, subject to $\|\boldsymbol{X}\| = 1$, can be obtained by reshaping into matrix format the eigenvector associated with the minimum eigenvalue of the semidefinite positive Hermitian matrix $\boldsymbol{T}^*\boldsymbol{T}$, say $\boldsymbol{u}$, i.e., $\boldsymbol{X} = \mathrm{vec}^{-1}\,(\boldsymbol{u})$.

Let $\boldsymbol{U} = [\,\boldsymbol{U}_1\, \cdots\, \boldsymbol{U}_P\,]$. Then, $\boldsymbol{U} = \boldsymbol{Q}$, up to a phase offset per user.

Step 3. Defining

$$\widetilde{\boldsymbol{H}} = \boldsymbol{G}_0 \boldsymbol{U} \boldsymbol{R}_{\boldsymbol{s}}(0)^{-1/2},$$

and, by what has been proved so far, we have

$$\forall \tau \in \mathcal{T}: \qquad \boldsymbol{R}(\tau) = \widetilde{\boldsymbol{H}} \boldsymbol{R}_{\boldsymbol{s}}(\tau) \widetilde{\boldsymbol{H}}^H,$$

or, equivalently,

$$\widetilde{\boldsymbol{H}} = \boldsymbol{H},$$

up to a phase offset per user.

4 STOCHASTIC ML METHODS

Most of the maximum–likelihood (ML) approaches to the source separation problem assume deterministic source signals in white Gaussian noise. In this case, the problem is reduced to a least squares optimization

$$\left(\widehat{\boldsymbol{H}}, \widehat{\boldsymbol{S}}\right) = \arg\min_{\boldsymbol{H},\boldsymbol{S}} \|\boldsymbol{X} - \boldsymbol{H}\boldsymbol{S}\|_F^2 ,$$

whose solution can be found iteratively using some alternating minimization procedure, such as, e.g., the ILSP and the ILSE algorithms. Here, we use a stochastic maximum–likelihood (SML) approach, in the sense that we consider a random model for the source data sequences. We discuss the approach only in the context of instantaneous mixtures. The case of convolutive mixtures is more complex and, up to our knowledge, remains unstudied.

4.1 INSTANTANEOUS MIXTURES

We consider the same vector data model

$$\boldsymbol{x}(k) = \boldsymbol{H}\,\boldsymbol{s}(k) + \boldsymbol{n}(k), \quad k = 1, \ldots, K, \tag{117}$$

under the usual assumptions: the channel matrix $\boldsymbol{H}$ is assumed time invariant along the K snapshots; the noise vector $\boldsymbol{n}(k)$ is stationary and white in the time domain, having probability density function $p_{\boldsymbol{n}}(\boldsymbol{n})$; the noise and signal vectors are statistically independent; in eq. (117), each element $s_p(k)$ is a sample of a binary random variable that takes values in $\mathcal{A} = \{\pm 1\}$ with equal probability; the P binary data sequences generated by the P users are assumed white and statistically independent. Therefore, the signal vector takes values in the alphabet $\mathcal{A}^P$ which has cardinality $\mathcal{C}_{\mathcal{A}} = 2^P$.

To reconstruct the vector sequence $\{\boldsymbol{s}(k)\}_{k=1}^{K}$, we solve the multiple ($\mathcal{C}_{\mathcal{A}}-$ary) hypothesis test problem

$$H_i: \ \boldsymbol{x}(k) = \boldsymbol{H}\,\boldsymbol{s}_i + \boldsymbol{n}(k), \ \ \boldsymbol{s}_i \in \mathcal{A}^P, \ \ k = 1, \ldots, K. \tag{118}$$

Clearly, the vector $\boldsymbol{x}(k)$ represents the array data at snapshot k given that $\boldsymbol{s}_i \in \mathcal{A}^P$ was transmitted. In eq. (118), the channel matrix $\boldsymbol{H}$ is unknown. To solve this hypothesis detection problem with unknown parameters, we follow a generalized maximum–likelihood (GML) approach: we first determine the Maximum Likelihood (ML) estimate of $\boldsymbol{H}$ using the K array data snapshots, and then solve the multiple hypothesis test problem at each snapshot by replacing the unknown $\boldsymbol{H}$ by its ML estimate $\widehat{\boldsymbol{H}}$.

The conditional array data probability density function at time instant k given $\boldsymbol{H}$ and given that $\boldsymbol{s}(k) = \boldsymbol{s}_i \in \mathcal{A}^P$ was transmitted is

$$p_{\boldsymbol{x}|\boldsymbol{H},\boldsymbol{s}_i}(\boldsymbol{x}(k)) = p_{\boldsymbol{n}}(\boldsymbol{x}(k) - \boldsymbol{H}\,\boldsymbol{s}_i), \tag{119}$$

therefore the conditional probability density function of the array data at time k given $\boldsymbol{H}$ is

$$\begin{aligned} p_{\boldsymbol{x}|\boldsymbol{H}}(\boldsymbol{x}(k)) &= \textstyle\sum_{i=1}^{\mathcal{C}_{\mathcal{A}}} p_{\boldsymbol{x}|\boldsymbol{H},\boldsymbol{s}_i}(\boldsymbol{x}(k)) \Pr\{\boldsymbol{s}(k) = \boldsymbol{s}_i\} \\ &= 2^{-P} \textstyle\sum_{i=1}^{\mathcal{C}_{\mathcal{A}}} p_{\boldsymbol{n}}(\boldsymbol{x}(k) - \boldsymbol{H}\,\boldsymbol{s}_i). \end{aligned} \tag{120}$$

This is a finite linear mixture model. Given the K array snapshots, we write the generalized array data likelihood function

$$\Lambda\left([\boldsymbol{x}(1)\cdots\boldsymbol{x}(K)]\,|\boldsymbol{H}\right) = 2^{-PK} \prod_{k=1}^{K} \sum_{i=1}^{\mathcal{C}_{\mathcal{A}}} p_{\boldsymbol{n}}(\boldsymbol{x}(k) - \boldsymbol{H}\,\boldsymbol{s}_i), \tag{121}$$

or the equivalent log–likelihood function

$$\ln\Lambda\left([\boldsymbol{x}(1)\cdots\boldsymbol{x}(K)]\,|\boldsymbol{H}\right) = -PK\ln 2 + \sum_{k=1}^{K} \ln\left(\sum_{i=1}^{\mathcal{C}_{\mathcal{A}}} p_{\boldsymbol{n}}(\boldsymbol{x}(k) - \boldsymbol{H}\,\boldsymbol{s}_i)\right). \tag{122}$$

The ML estimate $\widehat{\boldsymbol{H}}$ of the channel mixing matrix $\boldsymbol{H}$ is

$$\widehat{\boldsymbol{H}} = \arg\max_{\boldsymbol{H}} \sum_{k=1}^{K} \ln\left(\sum_{i=1}^{\mathcal{C}_{\mathcal{A}}} p_{\boldsymbol{n}}(\boldsymbol{x}(k) - \boldsymbol{H}\,\boldsymbol{s}_i)\right). \tag{123}$$

Once the estimate $\widehat{\boldsymbol{H}}$ is obtained, we decide on the signal vector $\boldsymbol{s}(k) \in \mathcal{A}^P$ at each snapshot $k = 1, \cdots, K$, based on the *maximum a posteriori* (MAP) criterion:

$$\widehat{\boldsymbol{s}}(k) = \arg\max_{\boldsymbol{s}_i \in \mathcal{A}^P} \ \Pr\left\{\boldsymbol{s}(k) = \boldsymbol{s}_i | \boldsymbol{x}(k), \widehat{\boldsymbol{H}}\right\}, \ \ k = 1, \ldots, K, \tag{124}$$

where the a posteriori conditional probabilities

$$\Pr\left\{\boldsymbol{s}(k)=\boldsymbol{s}_i|\boldsymbol{x}(k),\widehat{\boldsymbol{H}}\right\}=\frac{p_{\boldsymbol{x}|\widehat{\boldsymbol{H}},\boldsymbol{s}_i}(\boldsymbol{x}(k))2^{-P}}{\sum_{i=1}^{\mathcal{C}_{\mathcal{A}}}\ p_{\boldsymbol{x}|\widehat{\boldsymbol{H}},\boldsymbol{s}_i}(\boldsymbol{x}(k))2^{-P}},\ \boldsymbol{s}_i\in\mathcal{A}^P, \tag{125}$$

or, tacking into account eq.(119),

$$\widehat{p}_i(\boldsymbol{x}(k),\widehat{\boldsymbol{H}})\doteq\Pr\left\{\boldsymbol{s}(k)=\boldsymbol{s}_i|\boldsymbol{x}(k),\widehat{\boldsymbol{H}}\right\}=\frac{p_{\boldsymbol{n}}(\boldsymbol{x}(k)-\widehat{\boldsymbol{H}}\boldsymbol{s}_i)}{\sum_{i=1}^{\mathcal{C}_{\mathcal{A}}}\ p_{\boldsymbol{n}}(\boldsymbol{x}(k)-\widehat{\boldsymbol{H}}\boldsymbol{s}_i)},\ \boldsymbol{s}_i\in\mathcal{A}^P. \tag{126}$$

Estimation of $\widehat{\boldsymbol{H}}$. The strong non–linearity of the log–likelihood function precludes the closed-form solution to the optimization in eq. (123). In [3, 4], this optimization problem is solved using the *Expectation–Maximization* (EM) algorithm [36]. When properly initialized, the EM algorithm converges in a few number of iterations to the actual solution. For the case under discussion here, and under the Gaussian noise assumption, the solution is obtained through the iteration

$$\widehat{\boldsymbol{H}}_{l+1}=\left(\sum_{k=1}^{K}\boldsymbol{x}(k)\frac{\sum_{i=1}^{\mathcal{C}_{\mathcal{A}}}p_{\boldsymbol{n}}(\boldsymbol{x}(k)-\widehat{\boldsymbol{H}}_l\boldsymbol{s}_i)\boldsymbol{s}_i^H}{\sum_{i=1}^{\mathcal{C}_{\mathcal{A}}}p_{\boldsymbol{n}}(\boldsymbol{x}(k)-\widehat{\boldsymbol{H}}_l\boldsymbol{s}_i)}\right)\left(\sum_{k=1}^{K}\frac{\sum_{i=1}^{\mathcal{C}_{\mathcal{A}}}p_{\boldsymbol{n}}(\boldsymbol{x}(k)-\widehat{\boldsymbol{H}}_l\boldsymbol{s}_i)\boldsymbol{s}_i\boldsymbol{s}_i^H}{\sum_{i=1}^{\mathcal{C}_{\mathcal{A}}}p_{\boldsymbol{n}}(\boldsymbol{x}(k)-\widehat{\boldsymbol{H}}_l\boldsymbol{s}_i)}\right)^{-1}, \tag{127}$$

or, using a more compact notation,

$$\widehat{\boldsymbol{H}}_{l+1}=\left(\sum_{k=1}^{K}\boldsymbol{x}(k)\sum_{i=1}^{\mathcal{C}_{\mathcal{A}}}\widehat{p}_i(\boldsymbol{x}(k),\widehat{\boldsymbol{H}}_l)\boldsymbol{s}_i^H\right)\ \left(\sum_{k=1}^{K}\sum_{i=1}^{\mathcal{C}_{\mathcal{A}}}\widehat{p}_i(\boldsymbol{x}(k),\widehat{\boldsymbol{H}}_l)\boldsymbol{s}_i\boldsymbol{s}_i^H\right)^{-1}. \tag{128}$$

It is interesting to notice that (128) is an iterative "right inversion" of

$$\boldsymbol{x}(k)=\boldsymbol{H}\boldsymbol{s}(k),\ k=1,\ldots,K.$$

If all the $\boldsymbol{s}(k),\ k=1,\ldots,K$ were known, then the inversion would be of the form

$$\left(\sum_{k=1}^{K}\ \boldsymbol{x}(k)\boldsymbol{s}^H(k)\right)\left(\sum_{k=1}^{K}\ \boldsymbol{s}(k)\boldsymbol{s}^H(k)\right)^{-1}\ =$$

$$\boldsymbol{H}\quad\left(\sum_{k=1}^{K}\ \boldsymbol{s}(k)\boldsymbol{s}^H(k)\right)\left(\sum_{k=1}^{K}\ \boldsymbol{s}(k)\boldsymbol{s}^H(k)\right)^{-1}=\boldsymbol{H}, \tag{129}$$

assuming that the inverse exists. Looking at eq.(128), we see that it can be written in a similar way as in (129). In fact, defining the a posteriori conditional expected value

$$E_{\boldsymbol{x},\widehat{\boldsymbol{H}}_l}\left\{\boldsymbol{s}(k)\right\}=\sum_{i=1}^{\mathcal{C}_{\mathcal{A}}}\ \widehat{p}_i(\boldsymbol{x}(k),\widehat{\boldsymbol{H}}_l)\boldsymbol{s}_i$$

and covariance

$$E_{\boldsymbol{x},\widehat{\boldsymbol{H}}_l}\left\{\boldsymbol{s}(k)\boldsymbol{s}^H(k)\right\}=\sum_{i=1}^{\mathcal{C}_{\mathcal{A}}}\ \widehat{p}_i(\boldsymbol{x}(k),\widehat{\boldsymbol{H}}_l)\boldsymbol{s}_i\boldsymbol{s}_i^H,$$

at iteration l, we have from eq. (128),

$$\widehat{\boldsymbol{H}}_{l+1}=\left(\sum_{k=1}^{K}\ \boldsymbol{x}(k)E_{\boldsymbol{x},\widehat{\boldsymbol{H}}_l}\left\{\boldsymbol{s}(k)\right\}\right)\ \left(\sum_{k=1}^{K}\ E_{\boldsymbol{x},\widehat{\boldsymbol{H}}_l}\left\{\boldsymbol{s}(k)\boldsymbol{s}^H(k)\right\}\right)^{-1}. \tag{130}$$

This clarifies the operation of the EM algorithm. The right inversion in the left hand side of (129) performs the maximization of the log–likelihood function and corresponds to the maximization step of EM. Since the two factors involved depend on the unknown $\boldsymbol{s}(k),\ k = 1, \ldots, K$, the corresponding a posteriori conditional expected value and covariance estimates are used in (130). This corresponds to the E step of EM. The iteration proceeds until some stopping condition is satisfied.

Reconstruction of the sources' signals. When convergence of the EM algorithm is achieved, then $\widehat{\boldsymbol{H}} \simeq \widehat{\boldsymbol{H}}_{l+1} \simeq \widehat{\boldsymbol{H}}_l$. This means that the a posteriori conditional probabilities are immediately available

$$\widehat{p}_i(\boldsymbol{x}(k), \widehat{\boldsymbol{H}}) \simeq \widehat{p}_i(\boldsymbol{x}(k), \widehat{\boldsymbol{H}}_l),$$

and the reconstruction of the data transmitted by the sources can be done using the MAP solution given in eq. (124).

In [4] it is shown through extensive computer simulations that, even at moderate signal to noise ratios, the EM algorithm converges to the optimal ML solution in a few number of iterations and tracks satisfactorily the channel changes. Due to its highly parallelizable capability EM can be implemented with computational costs that compare favorably with other algorithms.

5 SOURCE SEPARATION BY MODEL GEOMETRIC PROPERTIES

In this section, we describe new techniques to achieve source separation that exploit specific geometric features of the data model. The first technique, introduced in subsection 5.1, is restricted to high SNR scenarios and binary sources, with few data samples available. The small number of observations precludes the use of SOS. A geometrical convex re-formulation of the blind source separation problem is exploited to resolve the linear mixture of binary users. In subsection 5.2, we discuss a SOS-based semi-blind ML technique for channel identification with white (up to 2nd order) stationary inputs. The likelihood function for the residual unitary matrix is optimized directly over the manifold of orthogonal matrices, by a geodesic descent algorithm.

5.1 DETERMINISTIC SEPARATION: POLYHEDRAL SETS

Recall the data model in eq. (65), and let $s_p(k)$ denote binary signals, i.e., $s_p(k) = \pm 1$, for $p = 1, \ldots, P$ and $k = 1, \ldots, K$. For clarity, consider noiseless samples, i.e., $\boldsymbol{n}(k) = \boldsymbol{0}$, and let all parameters in the data model be real (no complex data). Moreover, assume $\boldsymbol{H}$ is a $P \times P$ non-singular matrix; this entails no loss of generality – recall the subspace method in subsection 2.2, and eq. (64). Our goal is to find equalizers $\boldsymbol{\xi} \in \mathbb{R}^P$ that, when applied to the observed data $\boldsymbol{x}(k)$, extract a binary user from the linear

mixture, i.e., we seek vectors in

$$\mathcal{E} = \left\{ \boldsymbol{\xi} \in \mathbb{R}^P \ : \ \boldsymbol{\xi}^T \boldsymbol{x}(k) = \pm s_p(k), \text{ for some } p = 1, \ldots, P \text{ and all } k = 1, \ldots, K \right\}. \tag{131}$$

In the sequel, we refer to $\mathcal{E}$ as the set of linear equalizers. Remark that $\boldsymbol{\xi} \in \mathcal{E}$ verifies the strict constant modulus equality

$$\text{for } k = 1, \ldots, K: \qquad |\boldsymbol{\xi}^T \boldsymbol{x}(k)| = 1, \tag{132}$$

Thus, $\boldsymbol{\xi} \in \mathcal{E}$ belongs to the data-dependent polyhedron

$$\mathcal{P} = \left\{ \boldsymbol{\xi} \in \mathbb{R}^P \ : \ |\boldsymbol{\xi}^T \boldsymbol{x}(k)| \leq 1, \text{ for all } k = 1, \ldots, K \right\}, \tag{133}$$

obtained by relaxing the equality in (132) to an inequality. In fact, the set $\mathcal{E}$ plays a special role within $\mathcal{P}$: it is exactly the set of its vertices (extreme points), under a certain condition stated in the theorem below.

Theorem 5.1 *Consider the signal model $\boldsymbol{x}(k) = \boldsymbol{H}\boldsymbol{s}(k)$, $k = 1, \ldots, K$, where the $P \times P$ channel matrix $\boldsymbol{H}$ is non-singular and suppose all distinct 2^P binary vectors of length P are represented in $\{\boldsymbol{s}(k) : k = 1, \ldots, K\}$. Then, the set of extreme points of the polyhedron $\mathcal{P}$ in (133) is given by $\mathcal{E}$ in (131).* □

The proof can be found in [48]. Thus, finding an equalizer in $\mathcal{E}$ is equivalent to finding a vertex of $\mathcal{P}$. To achieve this goal, we may exploit the fact that extreme points of a compact polyhedral set $\mathcal{S}$ are global minima of any linear function over $\mathcal{S}$. As a consequence, a $\boldsymbol{\xi} \in \mathcal{E}$ may be found by solving a linear programming (LP) problem

$$\boldsymbol{\xi} = \arg \min_{\boldsymbol{\omega} \in \mathcal{P}} \boldsymbol{c}^T \boldsymbol{\omega},$$

where $\boldsymbol{c}$ denotes a non-zero vector. To extract the remaining $P - 1$ equalizers, the following recursive approach may be pursued.

Suppose $1 \leq p < P$ equalizers, $\{\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_p\}$ have been retrieved. We seek a new vertex $\boldsymbol{\xi}_{p+1}$ of $\mathcal{P}$, i.e.,

$$\boldsymbol{\xi}_{p+1} \notin \mathcal{L}_p = \text{span}\left\{\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_p\right\}.$$

Recalling that the maximum of any convex function $f(\boldsymbol{\omega})$ over a compact polyhedral set $\mathcal{S}$ is attained at one of its extreme points, we can find $\boldsymbol{\xi}_{p+1}$ by solving a quadratic programming (QP) problem:

$$\boldsymbol{\xi}_{p+1} = \arg \min_{\boldsymbol{\omega} \in \mathcal{P}} \boldsymbol{\omega}^T \boldsymbol{U}_p \boldsymbol{\Lambda} \boldsymbol{U}_p^T \boldsymbol{\omega},$$

where the $P \times (P - p)$ matrix $\boldsymbol{U}_p$ spans the orthogonal complement of $\mathcal{L}_p$ and the $(P - p) \times (P - p)$ matrix $\boldsymbol{\Lambda}$ denotes a positive definite diagonal matrix. The role of the convex function

$$f(\boldsymbol{\omega}) = \boldsymbol{\omega}^T \boldsymbol{U}_p \boldsymbol{\Lambda} \boldsymbol{U}_p^T \boldsymbol{\omega}$$

is to prevent convergence to the already retrieved equalizers $\boldsymbol{\xi}_i$, $i = 1, \ldots, p$. Notice that $f(\boldsymbol{\omega}) \geq 0$ with equality if and only if $\boldsymbol{\omega} \in \mathcal{L}_p$. Thus $f(\boldsymbol{\xi}_{p+1}) > 0$ implies that $\boldsymbol{\xi}_{p+1}$ is non-redundant, i.e., $\boldsymbol{\xi}_{p+1} \notin \mathcal{L}_p$.

In the presence of noise, the conditions of the *Theorem* 5.1 are not met, and not all extreme points in $\mathcal{P}$ are necessarily close to equalizers in $\mathcal{E}$. This implies that convergence of the above optimization algorithms to spurious minima (or maxima) has to be checked, in order to re-initialize them with a distinct starting point. Because of this, this simple geometrically-based methodology for blind separation of binary sources is only restricted to high SNR scenarios.

5.2 SEMI-BLIND ML SEPARATION OF WHITE SOURCES

We delineate a semi-blind technique [49] to identify the MIMO channel matrix $\boldsymbol{H}$: recall the data model in eq. (65) and assumptions (A1)-(A3). For simplicity, we also assume all data to be real. The technique is termed semi-blind because we assume that, for each source $p = 1, \ldots, P$, a certain fragment

$$\mathcal{F}_p(i_p, j_p) = \{s_p(i_p), s_p(i_p + 1), \ldots, s_p(j_p)\}$$

of the emitted message is known by the receiver, e.g., a portion of the frames' header. Notice that it is not required that $i_p = i_q$ (nor, $j_p = j_q$) for two distinct sources $p \neq q$, i.e., no synchronization among the sources is assumed, which would be difficult to satisfy in practice, but only between each source and the base station receiver.

The assumption of certain symbols being known is needed because as seen before and as it is well known, even in the noiseless case, the factorization $\boldsymbol{H}\boldsymbol{s}(k)$ is not unique: $\boldsymbol{H}$ can only be solved up to a residual $P \times P$ instantaneous mixing matrix [44, 15]. Notice further that, here, no finite-alphabet assumption is made on the sources, which is also a possible way to resolve the mixing ambiguity as seen in the above sections.

The technique works as follows. We assume that the data samples have been pre-whitened as explained in section 3.1. Thus, we find ourselves in the data model of (69), reproduced here for the convenience of the reader:

$$\boldsymbol{y}(k) = \boldsymbol{Q}\boldsymbol{s}(k) + \boldsymbol{w}(k), \quad k = 1, 2, \ldots, K, \tag{134}$$

where the corresponding (unknown) $PL \times PL$ channel matrix $\boldsymbol{Q}$ is unitary and has less parameters to estimate than the associated $N \times PL$ matrix $\boldsymbol{H}$. For the pth source, collect all the data samples that play a role in (134) in the vector $\boldsymbol{s}_p$, i.e.,

$$\boldsymbol{s}_p = (s_p(2 - L), s_p(3 - L), \ldots, s_p(K - 1), s_p(K))^T,$$

and stack these in the overall data vector

$$\boldsymbol{s} = \left[\boldsymbol{s}_1^T \; \boldsymbol{s}_2^T \; \cdots \; \boldsymbol{s}_P^T \right]^T,$$

For further reference, we express in matrix terms the knowledge of the data fragments as

$$\boldsymbol{E}_p^T \boldsymbol{s}_p = \boldsymbol{\eta}_p,$$

where the $(K-L+1) \times (j_p - i_p + 1)$ matrix $\boldsymbol{E}_p$ has a single 1 in each of its columns, will all the remaining entries being 0, that selects the *a priori* known entries in $\boldsymbol{s}_p$, and

$$\boldsymbol{\eta}_p = (s_p(i_p), s_p(i_p+1), \ldots, s_p(j_p))^T$$

contains the known data fragment. Thus, the overall knowledge for all the sources is expressed as

$$\begin{aligned} \boldsymbol{E}^T \boldsymbol{s} &= \boldsymbol{\eta} \\ \boldsymbol{E} &= \operatorname{diag}(\boldsymbol{E}_1, \ldots, \boldsymbol{E}_P) \\ \boldsymbol{\eta} &= \left[\boldsymbol{\eta}_1^T \cdots \boldsymbol{\eta}_P^T \right]^T. \end{aligned}$$

Since the statistical description of the sources is not known beyond the second order, we can say loosely that the pre-whitening step has exhausted all the stochastic information about them. Thus, in the sequel, $\boldsymbol{s}$ is treated as a deterministic vector of parameters; the entries of $\boldsymbol{Q}$ are also deterministic. The joint maximum-likelihood estimates of $(\boldsymbol{Q}, \boldsymbol{s})$, subject to the known constraints, is given by

$$\widehat{(\boldsymbol{Q}, \boldsymbol{s})} = \arg \max_{\boldsymbol{Q} \in \mathcal{O}, \boldsymbol{E}^T \boldsymbol{s} = \boldsymbol{\eta}} \Lambda(\boldsymbol{y}(1), \ldots, \boldsymbol{y}(K) \,|\, \boldsymbol{Q}, \boldsymbol{s}), \tag{135}$$

where

$$\mathcal{O} = \left\{ \boldsymbol{U} \in \mathbb{R}^{PL \times PL} \,:\, \boldsymbol{U}^T \boldsymbol{U} = \boldsymbol{I}_{PL} \right\}$$

denotes the set of $PL \times PL$ orthogonal matrices, and $\Lambda(\cdot | \boldsymbol{Q}, \boldsymbol{s})$ is the likelihood of the pre-whitened data samples $\boldsymbol{y}(k)$ conditioned on the pair $(\boldsymbol{Q}, \boldsymbol{s})$. Equation (135) is equivalent to

$$\widehat{(\boldsymbol{Q}, \boldsymbol{s})} = \arg \min_{\boldsymbol{Q} \in \mathcal{O}, \boldsymbol{E}^T \boldsymbol{s} = \boldsymbol{\eta}} \phi(\boldsymbol{Q}, \boldsymbol{s}), \tag{136}$$

where

$$\phi(\boldsymbol{Q}, \boldsymbol{s}) = \frac{1}{K} \left\| \boldsymbol{Y} - \boldsymbol{Q} \boldsymbol{\mathcal{T}}(\boldsymbol{s}) \right\|_{C^{-1}}^2. \tag{137}$$

Here, $\boldsymbol{Y} = [\, \boldsymbol{y}(1)\, \boldsymbol{y}(2) \cdots \boldsymbol{y}(K) \,]$ contains the whitened data and

$$\boldsymbol{\mathcal{T}}(\boldsymbol{s}) = \begin{bmatrix} \boldsymbol{\mathcal{T}}(\boldsymbol{s}_1) \\ \boldsymbol{\mathcal{T}}(\boldsymbol{s}_2) \\ \vdots \\ \boldsymbol{\mathcal{T}}(\boldsymbol{s}_P) \end{bmatrix}$$

is a stack of P Toeplitz matrices, each given by

$$\boldsymbol{\mathcal{T}}(\boldsymbol{s}_p) = \begin{bmatrix} s_p(1) & s_p(2) & s_p(3) & \cdots & s_p(K) \\ s_p(0) & s_p(1) & \ddots & \ddots & s_p(K-1) \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ s_p(2-L) & s_p(3-L) & \cdots & \cdots & s_p(K-L+1) \end{bmatrix}.$$

Moreover, in eq. (137), $\boldsymbol{C}$ denotes the covariance matrix of the noise $\boldsymbol{w}(k)$, and, as usual,

$$\|\boldsymbol{Z}\|^2_{\boldsymbol{C}^{-1}} = \mathrm{tr}\left\{\boldsymbol{Z}^T\boldsymbol{C}^{-1}\boldsymbol{Z}\right\}.$$

The optimization problem expressed in eq. (137) has no apparent closed-form solution. However, an iterative cyclic coordinate descent approach may be employed. This is only locally convergent. The resulting algorithm, termed here as iterative maximum likelihood (IML), is given in table 1.

Let $\boldsymbol{Q}^{(0)} \in \mathcal{O}$
for $n = 1, 2, \ldots$
i) $\boldsymbol{s}^{(n)} = \arg\min_{\boldsymbol{E}^T\boldsymbol{s}=\boldsymbol{\eta}} \phi\left(\boldsymbol{Q}^{(n-1)}, \boldsymbol{s}\right)$
ii) $\boldsymbol{Q}^{(n)} = \arg\min_{\boldsymbol{Q}\in\mathcal{O}} \phi\left(\boldsymbol{Q}, \boldsymbol{s}^{(n)}\right)$
until $\boldsymbol{Q}^{(n)} - \boldsymbol{Q}^{(n-1)} = \boldsymbol{0}$

Table 1: IML Algorithm

Solving for $\boldsymbol{s}^{(n)}$ in substep (i) of the IML algorithm can be formulated as a standard Least Squares (LS) minimization problem with linear constraints, whose solution is well-known, see [49] for details. Substep (ii) is more involved and requires more attention. After some algebra, it can be seen that substep (ii) boils down to the minimization of

$$\varphi\left(\boldsymbol{Q}\right) = \mathrm{tr}\left\{\boldsymbol{Q}^T\boldsymbol{C}^{-1}\boldsymbol{Q}\widehat{\boldsymbol{R}}_{\boldsymbol{s}}\right\} - 2\,\mathrm{tr}\left\{\boldsymbol{Q}^T\boldsymbol{C}^{-1}\widehat{\boldsymbol{R}}_{\boldsymbol{ys}}\right\}, \tag{138}$$

where

$$\widehat{\boldsymbol{R}}_{\boldsymbol{s}} = \frac{1}{K}\boldsymbol{\mathcal{T}}\left(\boldsymbol{s}\right)\boldsymbol{\mathcal{T}}\left(\boldsymbol{s}\right)^T \text{ and } \widehat{\boldsymbol{R}}_{\boldsymbol{ys}} = \frac{1}{K}\boldsymbol{Y}\boldsymbol{\mathcal{T}}\left(\boldsymbol{s}\right)^T,$$

subject to $\boldsymbol{Q} \in \mathcal{O}$, the group of $PL \times PL$ orthogonal matrices. In order to exploit the curvature of this surface constraint a geodesic descent algorithm, which is a generalization of the traditional steepest gradient method in flat spaces [23] to curved manifolds, may be employed. This technique is described in table 2. Thus, the geodesic descent algorithm acts as classical line search methods, except for the fact that the lines are replaced by their counterparts in the constraint surfaces: the geodesics. Derivation of all the details of the geodesic descent algorithm (GDA) is out of our scope here, see [49]. A possible initialization for the GDA is given by

$$\boldsymbol{Q}^{(0)} = \boldsymbol{\Pi}_{\mathcal{O}}\left\{\boldsymbol{C}^{-1}\widehat{\boldsymbol{R}}_{\boldsymbol{y\theta}}\right\}, \tag{139}$$

where $\boldsymbol{\Pi}_{\mathcal{O}}\left(\boldsymbol{Z}\right)$ is the (nonlinear) projection of the matrix $\boldsymbol{Z} \in \mathbb{R}^{LP\times LP}$ onto the orthogonal group $\mathcal{O}$. It may be obtained as follows: let

$$\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$$

1. Choose $\boldsymbol{Q}^{(0)} \in \mathcal{O}(LP)$

2. for $m = 1, 2, \ldots$

   a) Let $\boldsymbol{D}$ denote the projection of $-\boldsymbol{\nabla}\varphi\left(\boldsymbol{Q}^{(m-1)}\right)$ onto the tangent space of $\mathcal{O}$ at $\boldsymbol{Q}^{(m-1)}$

   b) Let $\boldsymbol{Q}(t) \in \mathcal{O}$, $t \geq 0$, denote the geodesic emanating from $\boldsymbol{Q}(0) = \boldsymbol{Q}^{(m-1)}$ in the direction $\dot{\boldsymbol{Q}}(0) = \boldsymbol{D}$

   c) Minimize $\varphi\left(\boldsymbol{Q}(t)\right)$ with respect to $t \geq 0$, to obtain $t_{\min}$ (a global minimizer); set $\boldsymbol{Q}^{(m)} = \boldsymbol{Q}\left(t_{\min}\right)$

3. repeat 2. until $\boldsymbol{Q}^{(m)} - \boldsymbol{Q}^{(m-1)} = \boldsymbol{0}$

Table 2: Geodesic descent algorithm

denote a SVD of $\boldsymbol{Z}$; then,

$$\boldsymbol{\Pi}_{\mathcal{O}}\left(\boldsymbol{Z}\right) = \boldsymbol{U}\boldsymbol{V}^{T}.$$

The idea behind this initialization is that, near the global minimum,

$$\widehat{\boldsymbol{R}}_{\boldsymbol{s}} = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{s}\left(k\right)\boldsymbol{s}\left(k\right)^{T} \simeq \boldsymbol{R}_{\boldsymbol{s}} = \mathrm{E}\left\{\boldsymbol{s}\left(k\right)\boldsymbol{s}\left(k\right)^{T}\right\} = \boldsymbol{I}_{LP}.$$

Thus, the first term in eq. (138) reduces to a constant, and

$$\varphi\left(\boldsymbol{Q}\right) \simeq -2\,\mathrm{tr}\left\{\boldsymbol{Q}^{T}\boldsymbol{C}^{-1}\widehat{\boldsymbol{R}}_{\boldsymbol{ys}}\right\},$$

which is precisely minimized by eq. (139) [13].

Future research directions. The two geometrically inspired techniques for source separation presented in this section may be further developed. Future work on the polyhedral characterization of equalizers may include the extension of the proposed technique to higher cardinality digital modulation alphabets such as PAM, QAM, etc. Also, robustness to noise and a re-formulation of the concepts that we presented in order to avoid *a priori* channel length estimation are important topics for further study. With respect to the semi-blind ML approach, more efficient optimization techniques like Newton, or conjugate gradient, matched to the specific structure of the constraint manifolds at hand are to be investigated, in order to speed up the convergence of the proposed methods.

6 CONCLUSIONS

This chapter considers the blind channel identification and source separation problems in the context of SDMA wireless communication systems. We exploit array processing techniques for smart antennas technology. The approaches described did not rely on any type of spatial structure modeling, such as, array

and wavefront geometries. These features are assumed unknown and embedded in the Multiple-Input Multiple Output (MIMO) channel model. For both memoryless and ISI channels, the channel model is represented by an unknown channel matrix, referred to as instantaneous and convolutive mixing matrix, respectively. For memoryless channels, we provide solutions to the blind source separation problem. For MIMO–ISI channels, we study also the blind ISI cancellation (equalization) sub–problem. We organize the chapter according to the different approaches that we consider: deterministic methods, second order statistics (SOS) based techniques, and stochastic maximum likelihood (SML). Many of the known solutions to blind channel identification and source separation are implemented with iterative algorithms. This is the case with the ILSP and ILSE algorithms in section 2, the hypercube and the LS constellation derotator methods based on SOS described in section 3, and the SML–EM algorithm of section 4.We have also focused our attention on analytical and/or closed form solutions. This is a very important topic, since it avoids efficiently the global convergence problems of the iterative solutions. The ACMA and the linear coding methods for blind source separation, and the subspace method for ISI cancellation presented in section 2 are in the class of analytical or closed form deterministic solutions. Similarly, the OPDA, the TICC, and the CFC_2 approaches in section 3 provide closed form SOS solutions. With all these closed form solution algorithms, global convergence to the optimal solution comes with a price, namely, some increase in algorithmic computational complexity. We revisit this tradeoff between increased computational complexity for closed form solutions and global convergence of iterative methods in section 5 where we develop source separation algorithms that exploit distinct and specific geometric features of the data model. We present a deterministic algorithm in subsection 5.1 that is based on a geometrical convex formulation of the blind source separation problem. In subsection 5.2, we describe a SML approach to solve the semi–blind separation of white sources based on geodesic descent algorithms in specific manifolds related to the geometry of the data model. In both cases, global convergence is guaranteed. These two algorithms are the result of preliminary research on the application of convex and differential geometry driven optimization techniques to the blind source separation problem. We believe that this opens promising guidelines for future research.

References

[1] B. G. Agee, "Fast Adaptive Polarization Control Using the Least–Squares Constant Modulus Algorithm," in *Proc. 20th IEEE Asilomar Conference on Signals, Systems and Computers*, pp. 590–595, 1987.

[2] B. G. Agee, "Blind Separation and Capture of Communication Signals Using a Multitarget Constant Modulus Beamforming," in *Proc. IEEE MILCOM*, vol. 2, pp. 340–346, Boston, MA, 1989.

[3] V. A. N. Barroso and J. M. F. Moura, "Beamforming with Correlated Arrivals in Mobile Communications", in Vijay Madisetti and Douglas Williams, editors, *Digital Signal Processing Handbook* (invited paper), chapter on Array Signal Processing, edited by Mos Kaveh, CRC Press, 1997

[4] V. A. N. Barroso, J. M. F. Moura, and J. Xavier, "Blind Array Channel Division Multiple Access (AChDMA) for Mobile Communications", *IEEE Transactions on Signal Processing*, vol. 46, pp. 737–752, March 1998.

[5] R. T. Behrens and L. L. Scharf, "Signal Processing Aplications of Oblique Projection Operators," *IEEE Transactions on Signal Processing*, vol. 42, pp. 1413–1424, June 1994.

[6] A. Chevreuil and P. Loubaton, "MIMO Blind Second-Order Equalization Method and Conjugate Cyclostationarity," *IEEE Transactions on Signal Processing*, vol. 47, no. 2, pp. 572–578, February 1999.

[7] Z. Ding, "Matrix Outer-Product Decomposition Method for Blind Multiple Channel Identification," in *IEEE Transactions on Signal Processing*, vol. 45, no. 12, pp. 3053–3061, December 1997.

[8] Z. Ding, "Blind Wiener Filter Estimation for Multi-Channel Systems Based on Partial Information," in *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing–ICASSP*, pp. 3609–3612, Munich, Germany, 1997.

[9] G. B. Giannakis, "Filterbanks for Blind Channel Identification and Equalization," *IEEE Signal Processing Letters,* vol. 4, pp. 184-187, June 1997.

[10] D. N. Godard, "Self–Recovering Equalization and Carrier Tracking in Two Dimensional Data Communication Systems," in *IEEE Transactions on Communications*, vol. COM-28, pp. 1867–1875, November 1980.

[11] L. C. Godara, "Applications of Antenna Arrays to Mobile Communications, Part I: Performance Improvement, Feasibility, and System Considerations," in *Proceedings of the IEEE*, vol. 85, no. 7, pp. 1031–1060, July 1997.

[12] L. C. Godara, "Applications of Antenna Arrays to Mobile Communications, Part II: Beamforming and Direction–of–Arrival Considerations," in *Proceedings of the IEEE*, vol. 85, no. 8, pp. 1195–1245, August 1997.

[13] G. Golub and C. F. Van Loan, *Matrix Computations*, John Hopkins University Press, Baltimore and London, 1989.

[14] R. Gooch and J. Lundell, "The CM Array: An Adaptive Beamformer for Constant Modulus Signals," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing–ICASSP*,pp. 2523–2526, Tokyo, Japan, 1986.

[15] A. Gorokhov and P. Loubaton, "Subspace-Based Techniques for Blind Separation of Convolutive Mixtures with Temporally Correlated Sources", *IEEE Transactions on Circuits and Systems – I: Fundamental Theory and Applications*, vol. 44, no. 9, pp. 813–820, September 1997.

[16] A. Graham, *Kronecker Products and Matrix Calculus with Applications,* Ellis Horwood Series, Mathematics and Its Applications, 1981, England.

[17] L. Hansen and G. Xu, "A Fast Sequential Source Separation Algorithm for Digital Cochannel Signals," *IEEE Signal Processing Letters*, vol. 4, no. 2, pp. 58–61, February 1997.

[18] L. Hansen and G. Xu, "A Hyperplane-Based Algorithm for the Digital Co-channel Communications Problem," *IEEE Transactions on Information Theory*, vol. 43, no. 5, pp. 1536–1548, September 1997.

[19] A. V. Keerthi, A. Mathur, and J. J. Shync, "Direction Finding Performance of the Multistage CMA Array," in *Proc. 28th IEEE Asilomar Conference on Signals, Systems and Computers*, vol. 2, pp. 847–852, 1994.

[20] H. Krim and M. Viberg, "Two Decades of Array Signal Processing Research ," *IEEE Signal Processing Magazine*, pp. 67–94, July 1996.

[21] J. C. Liberti Jr. and T. S. Rappaport, *Smart Antennas for Wireless Communications: IS–95 and Third Generation CDMA Applications*, Prentice Hall Communications Engineering and Emerging Technologies Series, Ed. Theodore S. Rappaport, 1999.

[22] T. Li and N. D. Sidiropoulos, "Blind Separation of Linear Mixtures of Digital Signals Using Successive Interference Cancellation Iterative Least Squares," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing–ICASSP*, vol. 5, pp. 2703–2706, Phoenix, USA, March 1999.

[23] D. Luenberger, *Linear and Nonlinear Programming*, 2nd ed., Addison Wesley

[24] H. Lui and X.-G. Xia, "Precoding Techniques for Undersampled Multireceiver Communication Systems," *IEEE Transacations on Signal Processing,* vol. 48, no. 7, pp. 1853–1864, July 2000.

[25] K. A. Meraim, P. Loubaton, and E. Moulines, "A Subspace Algorithm for Certain Blind Identification Problems," *IEEE Transactions on Information Theory*, vol. 43, no. 2, pp. 499–511, March 1997.

[26] E. Moulines, P. Duhamel, J. Cardoso, and S. Mayrargue, "Subspace Methods for the Blind Identification of Multichannel FIR Filters", *IEEE Transactions on Signal Processing*, vol. 43, no. 2, pp. 516–525, February 1995.

[27] A. J. Paulraj and C. Papadias, "Space–Time Processing for Wireless Communications," in *IEEE Transactions on Personal Communications*, vol. 14, no. 5, pp. 49–83, November 1997.

[28] Y. Sato, "A Method for Self–Recovering Equalization for Multilevel Amplitude–Modulation Problems," in *IEEE Transactions on Communications*, vol. COM-23, pp. 679–682, June 1975.

[29] A. Scaglione, G. B. Giannakis, and S. Barbarossa, "Redundant Filterbank Precoders and Equalizers Part I: Unification and Optimal Designs," *IEEE Transaction on Signal Processing,* vol. 47, no. 7, pp. 1988–2006, July 1999.

[30] A. Scaglione, G. B. Giannakis, and S. Barbarossa, "Redundant Filterbank Precoders and Equalizers Part II: Blind Channel Estimation, Synchronization, and Direct Equalization," *IEEE Transaction on Signal Processing,* vol. 47, no. 7, pp. 2007–2022, July 1999.

[31] R. O. Schmidt, "Multiple Emitter Location and Signal Parameter Estimation," in *IEEE Transactions on Antennas and Propagation*, vol. AP-34, no. 3, pp. 276–280, March 86.

[32] J. J. Shync and R. P. Gooch, "Convergence Properties of the Multistage CMA Beamformer," in *Proc. 27th IEEE Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 622–626, 1993.

[33] S. Talwar, M. Viberg, and A. Paulraj, "Blind Estimation of Multiple Co-Channel Digital Signals Using an Antenna Array," in *IEEE Signal Processing Letters*, vol. 2, no. 1, pp. 29–31, February 1994.

[34] S. Talwar, M. Viberg, and A. Paulraj, "Blind Separation of Synchronous Co-Channel Digital Signals Using an Antenna Array - Part I: Algorithms," in *IEEE Transactions on Signal Processing*, vol. 44, no. 5, pp. 1184–1197, May 1996.

[35] S. Talwar and A. Paulraj, "Blind Separation of Synchronous Co-Channel Digital Signals Using an Antenna Array – Part II: Performance Analysis," in *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 706–718, March 1997.

[36] D. M. Titterington, A. F. M. Smith, and U. E. Markov, *Statistical Analysis of Finite Mixture Distributions*. New York: Wiley, 1985.

[37] L. Tong, G. Xu, and T. Kailath, "Blind Identification and Equalization Based On Second-Order Statistics: A Time Domain Approach," *IEEE Transactions on Information Theory,* vol. 40, no. 2, pp. 340–349, March 1994.

[38] J. R. Treichler and M. G. Larimore, "New Processing Techniques Based on Constant Modulus Adpative Algorithm," in *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-33, pp. 420–431, April 1985.

[39] J. K. Tugnait, "On Linear Predictors for MIMO Channels and Related Blind Identification and Equalization," *IEEE Signal Processing Letters*, vol. 5, pp. 289–291, November 1998.

[40] J. K. Tugnait and B. Huang, "Blind Channel Estimation and Equalization of Multiple-Input Multiple-Output Channels," in *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'99)*, vol. 5, pp. 2707–2710, 1999.

[41] J. K. Tugnait and B. Huang, "Multi-Step Linear Predictors-Based Blind Equalization of Multiple-Input Multiple-Output Channels," in *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'99)*, vol. 5, pp. 2949–2952, 1999.

[42] P. Vandaele, G. Leus, and M. Moonen, "A Non–Iterative Blind Binary Signal Separation Algorithm Based on Linear Coding," in *Proceedings IEEE International Workshop on Signal Processing Advances for Wireless Communications (SPAWC'99)*, pp. 98–101, Annapolis, USA, May 1999.

[43] B. D. Van Veen and K. M. Buckley, "Beam-Forming: A Versatile Approach to Spatial Filtering," in *IEEE Magazine on Signal Processing*,pp. 4–24, April 1988.

[44] A. J. van der Veen, S. Talwar, and A. Paulraj, "A Subspace Approach to Blind Space–Time Signal Processing for Wireless Communications Systems," in *IEEE Transactions on Signal Processing*, vol. 45, no. , pp. 173–190, January 1997.

[45] A. J. van der Veen and A. Paulraj, "An Analytical Constant Modulus Algorithm," in *IEEE Transactions on Signal Processing*, vol. 44, no. 5, pp. 1136–1155, May 1996.

[46] J. Xavier and V. Barroso, "Blind Source Separation, ISI Cancelation and Carrier Phase Recovery in SDMA Systems for Mobile Communications," *Wireless Personal Communications*, special issue on *Wireless Broadband Communications*, vol. 10, pp. 36-76, Kluwer Academic Publishers, June 1999.

[47] J. Xavier and V. Barroso, "Correlative Coding Approach for Blind Identification of MIMO Systems," in *Proc. 2nd IEEE Signal Processing Workshop on Signal Processing Advances in Wireless Communications (SPAWC'99)*, pp. 263–266, Annapolis, USA, May 1999.

[48] J. Xavier and V. Barroso, "Polyhedral Methods for Blind Deterministic Separation of Binary Co-Channel Signals," in *Proceedings Global Telecommunications Conference, Globecom'99*, pp. 2269–2272, December 1999.

[49] J. Xavier and V. Barroso, "Semi-blind Maximum Likelihood Separation of Linear Convolutive Mixtures," in *Proc. 10th IEEE Workshop on Statistical Signal and Array Processing (SSAP'2000)*, pp. 329–333, August 2000.

[50] J. Xavier, V. Barroso and J. M. F. Moura, "Closed-Form Blind Identification of MIMO Channels by Correlative Coding: an Isometry Fitting to Second Order Statistics," to appear in *IEEE Trans. on Signal Processing*.