Blind Paraunitary Equalization
Sylvie Icart, Pierre Comon, Ludwig Rota

To cite this version:

HAL Id: hal-00354719
https://hal.archives-ouvertes.fr/hal-00354719
Submitted on 20 Jan 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Blind Paraunitary Equalization

Sylvie Icart, Pierre Comon,
Laboratoire I3S, CNRS UMR 6070, Université de Nice, 2000 route des Lucioles, BP 121, 06903 Sophia-Antipolis Cedex France

Ludwig Rota
IMRA Europe, Sophia-Antipolis

Abstract

In this paper a blind MIMO space-time equalizer is described, dedicated to convolutive mixtures when observations have been pre-whitened. Filters preserving space-time whiteness are paraunitary; a parameterization of such filters is proposed. Theoretical developments then lead to a numerical algorithm that sweeps all pairs of delayed outputs. This algorithm involves the solution of a polynomial system, whose coefficients depend on the output cumulants. Simulations and performance of the numerical algorithm are reported.

Key words: Blind deconvolution, Blind equalization, Multiple-Input/Multiple-Output (MIMO), High Order Statistics (HOS), Paraunitary filter.

1. Introduction

In actual digital communication systems, the equalization problem is solved using learning sequences. These sequences, known by transmitter and receiver, permit to estimate channel parameters. Nevertheless, in the future, learning sequences may be seen as reducing the throughput, because they occupy a non negligible space in transmitted sequences. Hence less actual information is transmitted, i.e. useful data rate is lower than system data rate. Next, in some situations, e.g. interception in electronic warfare, learning sequences exist but are not known to the receiver.

Blind separation methods for Multiple-Input Multiple-Output (MIMO) channels have raised an increasing interest for digital communications since they do not need learning sequences. Most blind MIMO equalization techniques use High Order Statistics (HOS) for separating signals. This can be implicit through constant modulus or constant power criteria. Purely deterministic approaches also exist and exploit either the finite alphabet property or the presence of another diversity in addition to time and space.

Our main contribution consists of a block algorithm dedicated to blind MIMO equalization and exploiting HOS of observations. The particularity of our method is that it is based on a factorization of paraunitary filters, as suggested in [21]. The paraunitary condition is not very restrictive since prewhitening can always be performed in a first stage (in a non unique manner) even if it is not always an obvious operation. Moreover, the algorithm designed herein can be implemented either “on-line” or “off-line”. More precisely, our algorithm can run iterations on a single given block until convergence, or on the contrary, run a new iteration after each symbol arrival. Convergence of on-line algorithms is known to be much longer in terms of number of symbols required (typically sev-
eral thousands of symbols).

Note that algorithms dedicated to paraunitary channels as in [23] or PAJOD [22], have already been proposed. Unfortunately, the paraunitary constraint was not accurately verified in [23], especially at low SNR, because the channel was parametrized with a semi-unitary matrix.

The paper is organized as follows. In the second section, model and hypotheses are presented. The parameterization of MIMO paraunitary filters is introduced in Section 3, and input-output cumulant relations are described in the fifth section. The core of the algorithm built from previous theoretical results is described in the last section.

2. Model and notations

Throughout the paper, (\(\dagger\)) stands for transposition, (\(\ddagger\)) for conjugate transposition, (\(\ast\)) for complex conjugation, and \(j = \sqrt{-1}\). Also denote by \(\mathbb{Z}\) the set of integers, by \(\mathbb{N}\) the subset of positive integers and by \(z^{-1}\) the time-delay operator. Vectors and matrices are denoted with bold lowercase and bold uppercase letters respectively. The entries of matrix \(G\) are denoted \(G_{ij}\), where subscript \(ij\) denotes the \(i\)-th row and the \(j\)-th column of \(G\).

Let us consider a digital communication system of \(N\) antennas and \(N\) receivers in a multipath environment. Let \(s(n) = (s_1(n), \ldots, s_N(n))^T\) denote the \(N\) - dimensional source vector of baseband complex signals at time \(n\) and \(w(n) = (w_1(n), \ldots, w_N(n))^T\) the \(N\) - dimensional observation vector. Let \(C[z]\) be the transfer function of the Linear Time Invariant (LTI) mixing channel and \(\{C(k), k = 0, \ldots, K\}\) be the sequence of \(N \times N\) matrices of the complex Finite Impulse Response (FIR) of the channel. We then have:

\[
w(n) = \sum_{k=0}^{K} C(k) s(n - k)
\]

where \(K\) denotes the memory length of the channel. Also denote

\[
C[z] = \sum_{m=0}^{K} C(m) z^{-m}.
\]

The case of instantaneous mixtures is not considered in the present paper, since already addressed via pairwise processing by various authors since 1991. Thus, it will be assumed that \(K > 0\).

The multichannel blind deconvolution problem consists of finding a LTI filter \(H[z]\), i.e. the equalizer, in order to retrieve the \(N\) input signals \(s_i(n), i \in \{1, \ldots, N\}, \forall n \in \mathbb{Z}\), solely from the observation of the outputs \(w(n)\) of the unknown LTI channel \(C[z]\).

Let \(\hat{s}(n) = (\hat{s}_1(n), \ldots, \hat{s}_N(n))^T\) be the \(N\)-dimensional estimated source vector. This means, with the above notation:

\[
\hat{s}(n) = \sum_{l=0}^{L} H(l) w(n - l)
\]

where \(L\) is the memory length of the equalizer, or, as a function of original sources themselves:

\[
\hat{s}(n) = \sum_{m=0}^{K+L} G(m) s(n - m),
\]

where \(G[z]\) denotes the global filter \(G[z] = H[z] C[z]\) (see fig 1 below).

**Definition 1 Paraunitarity.** A square polynomial matrix \(H[z] \in \mathbb{C}^{N \times N}\) is said to be paraunitary [22] if

\[
H^*[1/z^*] H[z] = I_N = H[z] H^*[1/z^*]
\]

where \(I_N \in \mathbb{R}^{N \times N}\) is the identity matrix.

In this paper, we assume the following hypotheses:

(H1) Inputs \(s_i(n), \forall i \in \{1, \ldots, N\}, \forall n \in \mathbb{Z}\), are mutually independent and identically distributed (i.i.d.) zero-mean random processes, with unit variance.
Vector \( s(n) \) is stationary up to the considered order \( r, r \geq 3, \) i.e. \( \forall i \in \{1, \ldots, N\}, \) the order-\( r \) marginal cumulants,

\[
\mathcal{C}_r^p [s_i] = \text{Cum}(s_i(n), \ldots, s_i(n), s_i^*(n), \ldots, s_i^*(n))
\]

do not depend on \( n. \) For definitions of cumulants, refer to [23] and references therein.

Remark 2 Hypothesis [H4] is not restrictive. Indeed, one can always whiten the observations by using a filter that factorizes the second-order power spectrum, i.e. a classical prewhitening of the observations [23,22]. In other words, paraunitary filters are relevant after a space-time standardization of observations (second order white with unit covariance). The way space-time whitening is implemented is out of the scope of this paper, and it is assumed that observations \( w(n) \) are space-time white.

As is now well known, statistical independence of \( \hat{s}_i \) can only allow to blindly recover the sources up to a permutation matrix \( P, \) and a diagonal delay filter \( A[z], \) so that \( H[z]C[z] = A[z]P. \)

3. Equalization

We introduce in this section the global parameterization of the paraunitary equalizer, which will allow in Section 4 an implementation in terms of Givens rotations.

3.1. Parameterization

From our observation model and from hypotheses described in Section 4, we have a first proposition:

Proposition 3 Let \( H[z] \in \mathbb{C}^{N \times N} \) be a FIR paraunitary filter of MacMillan degree \( \lambda h. \) Then matrix \( H[z] \) can be factorized (non uniquely) as the product:

\[
H[z] = A[z]QB[z]
\]

where \( Q \in \mathbb{C}^{N \times N} \) is unitary and \( A[z] \) and \( B[z] \) are FIR paraunitary filters of MacMillan degree \( \ell_A \) and \( \ell_B \) respectively, with \( 0 \leq \ell_A \leq \lambda_h, \) \( 0 \leq \ell_B \leq \lambda_h, \) \( \ell_A + \ell_B = \lambda_h. \)

PROOF. The proof is trivial since we can extend the paraunitary factorization of [23] to the \( N \)-dimensional complex case:

\[
H[z] = Q_0 Z[z] \cdots Z[z] Q_{\lambda_h - 1} Z[z] Q_{\lambda_h}
\]

where \( Q_p \in \mathbb{C}^{N \times N} \) are unitary for \( p \in \{0, \ldots, \lambda_h\}, \) \( \lambda_h \) is the McMillan degree of \( H[z], \) and \( Z[z] \) denotes the \( N \times N \) diagonal matrix

\[
Z[z] = \begin{bmatrix}
I_{N-1} & 0 \\
0 & z^{-1}
\end{bmatrix}.
\]

Thus, for \( 0 < \ell_A \leq p, \) we have

\[
A[z] = Q_0 Z[z] \cdots Z[p-1] Z[z]
\]

and for \( 0 < \ell_B \leq \lambda_h - p, \) we have

\[
B[z] = Z[z] Q_{p+1} \cdots Z[z] Q_{\lambda_h}.
\]

When \( \ell_A = 0 \) (respectively \( \ell_B = 0, \) we can replace \( A[z] \) (respectively \( B[z] \)) by \( I_N. \) Then, for a fixed \( p, \) any paraunitary filter of of MacMillan degree \( \lambda_h \) can be factorized like in (9).

3.2. MIMO relations

In the remaining, we assume the following notation for fourth order cumulants, e.g. cumulants of vector \( w, \)

\[
\Gamma_{e g, f h}(\nu) = \text{Cum}[w_e(n - \nu), w_g(n - \nu), w_f(n - \nu), w_h(n - \nu)]
\]

where \( e, f, g, h \) take their values in \( \{1, \ldots, N\}, \) and \( \nu_i \in \mathbb{N}, \forall i \in \{1, \ldots, 4\}. \)

Now, consider the following input-output relations for the convolutive model

\[
\hat{s}_i(n) = \sum_{m=0}^{l_A} \sum_{r=1}^{N} A_{iq}(m) Q_{qr} x_r(n-m),
\]

and

\[
x_r(n-m) = \sum_{l=0}^{l_B} \sum_{s=1}^{N} B_{rs}(l) w_s(n-m-l). \]

From [13] and thanks to the multilinearity property of cumulants, we can express the input-output relations between cumulants of input \( x \) and output \( \hat{s}, \)

\[
\Gamma_{ik,jl}^{\hat{s}} = \sum_{abcde} \sum_{qrst} A_{ij}(\tau_1) A_{jk}(\tau_2) A_{kl}(\tau_3) Q_{qa} Q_{qb} Q_{qc} Q_{qd} Q_{re} Q_{rf} Q_{sg} Q_{sh} Q_{st}
\]

The McMillan degree of a paraunitary filter is also the degree of its determinant [23, p. 737].
with $\tau = (\tau_1, \tau_2, \tau_3, \tau_4)$. Since $\tilde{s}$ is the output of filter $A[z]Q$, the range of each $\tau_i$ is $[0, \ldots, \ell_A]$ and all indices $a, b, c, d, i, j, k, l, q, r, s, t$ take their values in $\{1, \ldots, N\}$. Then, we can write a similar relation between cumulants of input $w$ and output $x$,

$$\Gamma_{ac, bd}^x(\tau) = \sum_{p, \rho, \phi, \psi} B_{ac}(\rho_1)B_{bd}^*(\rho_2)B_{cd}(\rho_3)B_{de}(\rho_4)\Gamma_{e, \phi}^w(\tau)$$

with $\rho = (\rho_1, \rho_2, \rho_3, \rho_4)$. Here, the range of each $\rho_i$ is $[0, \ldots, \ell_B]$ since inputs are filtered by $B[z]$.

3.3. Construction of $Q_p$

Matrices $Q_p$ involved in the paraunitary factorization are unitary. It is well known that any $N \times N$ unitary matrix can be written as a product of $M = N(N-1)/2$ Givens rotations [30, p. 215] $U(i, j, \theta, \phi)$ with $1 \leq i < j \leq N$ and

$$U(i, j, \theta, \phi) =
\begin{bmatrix}
I_{n_1} & : & 0 & : & 0 \\
\vdots & & \cos \theta & & \sin \theta e^{j \phi} & \vdots \\
0 & : & I_{n_2} & : & 0 \\
\vdots & & -\sin \theta e^{-j \phi} & & \cos \theta & \vdots \\
0 & : & 0 & : & I_{n_3}
\end{bmatrix}
$$

where $n_1 + n_2 + n_3 = N - 2$. These unitary rotations allow to develop a Jacobi-type iteration algorithm for equalizer [8].

Since we use Givens matrices, we can make 2 remarks.

Remark 4 For $N = 2$, unitary matrix $Q_p$ consists of a single rotation

$$Q_p =
\begin{bmatrix}
\cos \theta_p & \sin \theta_p e^{j \phi_p} \\
-\sin \theta_p e^{-j \phi_p} & \cos \theta_p
\end{bmatrix}.
$$

Then, only one pair of angles $(\theta_p, \phi_p)$ is needed for each $Q_p$. Hence, when more than two signals are observed, it is also possible to use a deflation-type algorithm [7].

Remark 5 Proposition 3 tells that $\lambda_1 + 1$ unitary matrices are used to obtain a complete factorization of the equalizer. For instance, the $2 \times 2$ length-1 paraunitary filter defined by diag$(H[z]) = [-z^{-1}, z^{-1}]$ and 0 elsewhere, needs $\lambda_1 = \deg(\det(H[z])) = 2$ diagonal matrices $Z[z]$. Indeed, $H[z]$ has the following factorization

$$H[z] = Q_2 Z[z] Q_1 Z[z] e^{j \frac{\pi}{2}}
$$

where $Q_1 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ and $Q_2 = \begin{bmatrix} \tau_1 & \tau_2 \\ \tau_3 & \tau_4 \end{bmatrix}$.

The exponential is part of the unknown matrix $A[z]P$, an indetermination inherent in the problem dealt with in Section 3. It is also true for $L \geq 1$ since we can group all exponentials in a single matrix $A[z]P$. Thus, reflections, e.g. diag$(Q) = [1, -1]$, are covered by the parameterization of Proposition 3, thanks to the inherent indeterminacy. The parameterization (3) is thus complete.

4. Optimization criterion

Let us start this section with two key definitions. The first definition deals with the set of filters that do not affect the independence of sources, a key point of blind source separation methods. The second definition describes the main properties of a contrast, i.e. an optimization criterion for separating signals.

Definition 6 Trivial filters. The set $S$ of source processes is characterized by assumptions, such as

$$H_1$$

One defines the set $T$ of trivial filters, as containing all filters that do not affect these assumptions. In other words, $S$ is stable by the operation of $T$. For instance, filters of the form $A[z]P$, where $P$ is a permutation matrix, and $A[z]$ a diagonal filter, do not affect mutual independence between components of $s(n)$. If in addition $s(n)$ is an i.i.d. non Gaussian process, $A[z]$ should contain only pure delays, integer multiples of the sampling period, and fixed complex factors; in other words, the entries of $A[z]$ are of the form $\gamma e^{j \phi}$, with $k \in \mathbb{Z}$ and $|\gamma| = 1$.

Definition 7 Contrast. Let $H$ be a set of filters, and denote $H \cdot S$ the set of standardized linear processes obtained by operation of filters of $H$ on processes of $S$. Define $I$ the identity matrix. An optimization criterion, $\Upsilon(H; w)$, will be referred to as a contrast for $H \in H \cdot S$, if it satisfies the three properties below $\Upsilon$:

- Invariance: The contrast should not change within the set of acceptable solutions, which means that $\forall w \in H \cdot S, H \in T$ then $\Upsilon(H; w) = \Upsilon(I; w)$.
- Domination: If sources are already separated, any filter should decrease the contrast i.e. $\forall w \in S, H \in H$, then $\Upsilon(H; w) \leq \Upsilon(I; w)$.
Discrimination: The maximum contrast should be reached only for filters linked to each other via trivial filters: \( \forall w \in S, \Upsilon(H; w) = \Upsilon(I; w) \Rightarrow H \in T \).

Now, let us focus on the contrast used to carry out the equalization of the system. We can separate signals blindly by using an approach based on High Order Statistics. Indeed, since fourth-order cross-cumulants are a good measure of statistical independence, we can build a contrast based on these values

**Proposition 8** Consider sources having kurtosis of the same sign, \( \epsilon = \pm 1 \). Then, the source separation can be performed solely from channel outputs by maximizing the following contrast

\[
\Upsilon_{1,4} = \epsilon \sum_{i=1}^{N} \Gamma_{ii,ii}^\hat{s}
\]

where cumulants of \( \hat{s} \) are defined by

\[
\Gamma_{ii,ii}^\hat{s} = \sum_{abcd} \sum_{\tau} A_{qg}(\tau_1) A_{ir}^*(\tau_2) A_{is}(\tau_3) A_{il}^*(\tau_4) Q_{qa} Q_{gb}^* Q_{rb} Q_{ic}^* \tau_{ac,bd}(\tau).
\]

Hence, in order to estimate \( H[z] \), the optimization problem can be rewritten as:

\[
H = \operatorname{Arg\,max}_Q \Upsilon_{1,4}
\]

where \( Q \) stands for the set of all \( Q_p \).

It has been proved in 2 that \( \Upsilon_{1,4} \) is a contrast. We maximize \( \Upsilon_{1,4} \) with respect to each pair \((\theta, \phi)\) in turn. The sequence of values of \( \Upsilon_{1,4} \) obtained this way is monotonically increasing. Since it is also bounded above, it converges. Figure 2 is an example of such a behaviour.

### 5. A Jacobi-type algorithm

In this section we detail an iteration of the algorithm for one pair \((\theta, \phi)\) of Givens rotation. Then, as in Jacobi sweeping algorithms for matrices, we sweep \( M = N(N-1)/2 \) pairs for each unitary matrix \( Q_p \).

#### 5.1. Processing a pair of outputs

For the sake of clarity, we drop index \( p \). We have to find all pairs \((\theta, \phi)\) that maximize (14), the other pairs being fixed. To reach this goal, (20) is simplified firstly by expanding it, and secondly by collecting terms involving \( \theta \) or \( \phi \). In this manner, we obtain

\[
\Gamma_{ii,ii}^\hat{s} = \sum_{\tau} A_{qg}(\tau_1) A_{ir}^*(\tau_2) A_{is}(\tau_3) A_{il}^*(\tau_4) Q_{qa} Q_{gb}^* Q_{rb} Q_{ic}^* \tau_{ac,bd}(\tau).
\]

According to (13), (20) and (23), each coefficient \( K_{(a)}^{(n)} \) depends on indices \( a, b, c, d \) and \( q, r, s, t \) and as the product \( Q_{qa} Q_{gb}^* Q_{rb} Q_{ic}^* \) gives an expression of the form \( \cos^\alpha \theta \sin^\beta \phi \) with \( \alpha \in \{0, \ldots, 4\} \) and \( \eta \in \{\pm 1, \pm 2, \pm 3, \pm 4\}; \) let \( \eta = 2\beta + \alpha - 4 \), we get

\[
\Gamma_{ii,ii}^\hat{s} = \sum_{\alpha=0}^{4} \sum_{\beta=0}^{4-\alpha} K_{(a)}^{(2\beta+\alpha-4)} \cos^\alpha \theta \sin^\beta \phi \tau_{ac,bd}(\tau).
\]

**Remark 9** Computing each \( \Gamma_{ii,ii}^\hat{s} \) without simplifications require a total of \((L+1)^2 N^8\) loops (equation 21). Thus, readers can find an implementation trick in Appendix 3 for reducing this number of loops to \((L+1)^4 N^4\). Moreover, the method detailed in Appendix 3 uses less memory for storing matrices. Only the cumulant tensor \( \Gamma_{ac,bd}(\tau) \), which
is computed at the beginning of each loop, needs a large memory, i.e. up to $N^4(L + 1)^4$ values to store.

Next, we make the change of variables: \(\phi = \tan(t/2)\), with \(t = \tan\frac{\phi}{2}\), and \(\cos = \frac{e^{i\phi}}{1+e^{i\phi}}\), \(\sin = \frac{e^{-i\phi}}{1+e^{i\phi}}\) with \(u = \tan\theta\). Thus, the numerator and the denominator of the polynomial obtained are both of degree 8 in variable \(t\) and 4 in variable \(u\). For maximizing contrast, \(\frac{\partial T_{1,4}}{\partial t}\), we have to find roots of polynomial system \((25)\), i.e. stationary points of \(T_{1,4}\), obtained by canceling the partial derivatives:

\[
\begin{align*}
\Phi_1(u, t) &\triangleq \frac{\partial T_{1,4}}{\partial u} \\
\Phi_2(u, t) &\triangleq \frac{\partial T_{1,4}}{\partial t}
\end{align*}
\]

After some simplifications, e.g. obvious root \(u = 0\) in \(\Phi_2(u, t)\), polynomials obtained have different global degrees: degree 12 for \(\Phi_1(u, t)\) (leading monomial is \(t^8u^4\)), and degree 11 for \(\Phi_2(u, t)\) (leading monomial is \(t^3u^3\)). Then we consider only variable \(u\) for \(\Phi_1(u, t)\) and \(\Phi_2(u, t)\), and we collect terms of same degree in \(u\). Hence, system \((25)\) can be rewritten

\[
\begin{align*}
\Phi_1(u, t) &= \sum_{k=0}^4 \chi_{4-k}(t) u^k \\
\Phi_2(u, t) &= \sum_{k=0}^3 \xi_{3-k}(t) u^k
\end{align*}
\]

The polynomial system \((25)\) admits a solution if and only if the resultant, i.e. determinant of a Sylvester matrix, is null \((22)\). Thus, we obtain the determinant to solve

\[
\begin{vmatrix}
\xi_0(t) & 0 & 0 & 0 & \chi_0(t) & 0 & 0 \\
\xi_1(t) & \xi_0(t) & 0 & 0 & \chi_1(t) & \chi_0(t) & 0 \\
\xi_2(t) & \xi_1(t) & \xi_0(t) & 0 & \chi_2(t) & \chi_1(t) & \chi_0(t) \\
\xi_3(t) & \xi_2(t) & \xi_1(t) & \xi_0(t) & \chi_3(t) & \chi_2(t) & \chi_1(t) & \chi_0(t) \\
0 & \xi_3(t) & \xi_2(t) & \chi_4(t) & \chi_3(t) & \chi_2(t) & \chi_1(t) & \chi_0(t) \\
0 & 0 & \xi_3(t) & \xi_2(t) & \chi_4(t) & \chi_3(t) & \chi_2(t) & \chi_1(t) & \chi_0(t) \\
0 & 0 & 0 & \xi_3(t) & \chi_4(t) & \chi_3(t) & \chi_2(t) & \chi_1(t) & \chi_0(t)
\end{vmatrix} = 0. (27)
\]

It turns out that this polynomial generally admits no more than 16 real roots. When all real roots are found, we have to plug them back in \((22)\) for selecting the one leading to the absolute maximum of \((21)\). It is not that hard to compute since powerful algorithms exist that can rapidly find roots of univariate polynomials. The complete algorithm can be summarized as follows.

\begin{enumerate}
\item Compute \(\Gamma^{w, fh}_k(\tau + \rho)\)
\item Set \(M := N(N - 1)/2\) and \(T := \lfloor \sqrt{NM_h} \rfloor + 1\)
\item Initialize \(Q^{(k)}_h\) to identity (i.e all \((\theta, \phi)\) to zero) for all \(k\) and \(p\)
\item For each \(k\) do
\begin{enumerate}
\item For each \(p\) do
\begin{enumerate}
\item Compute \(A_p^{(k)}[z] = Q_{0}^{(k)} Z[z] \ldots Q_{p-1}^{(k)} Z[z] - Q_{p+1}^{(k-1)} Z[z] L_h^{\lambda_h}
\item Compute \(B_p^{(k)}[z] = Z[z] Q_{p+1}^{(k-1)} \ldots Z[z] L_h^{\lambda_h}\)
\item Compute \(\Gamma^{w, bh}_h(\tau)\) (with \(x^{(k)}[z] = B_p^{(k)}[z] w[z]\))
\item Search for \(Q_p^{(k)}\) maximizing \(T_{1,4}\)
\end{enumerate}
\end{enumerate}
\item Return \(H[z] = Q_0^{(T)} Z[z] \ldots Q_{p}^{(T)} Z[z] Q_{p+1}^{(T)} \ldots Z[z] L_h^{\lambda_h}\)
\end{enumerate}

5.2. Sweeping the pairs

Let us comment some points of this algorithm:
- the resulting tensor \(\Gamma^{w, bh}_h(\tau)\) is composed of \(N^2(L - \ell_B + 1)^2\) matrices each of size \(N(L - \ell_B + 1) \times N(L - \ell_B + 1)\). Hence, the algorithm needs to store \(N^4(L - \ell_B + 1)^4\) entries in memory. Simplifications are generally impossible since we need exact values for computing roots of polynomial system \((24)\).
- In order to increase the precision of the angles, we suggest to execute \(T = \lfloor \sqrt{N} \rfloor + 1\) sweeps. Actually, the first angles computed are not well defined since all other angles are still null (set at stage initialize). Hence, when loop on each \(k\) is repeated several times, angles are better estimated. This procedure reminds the well-known Jacobi sweeping widely used for eigenvalue computation.

Remark 10 In the case of a slowly time varying linear channel, the initialization step initialize can be modified since angles previously found give a better approximation than null values.

6. Computer Results

For large data blocks, the performance of the algorithm described in Section 3 comes close to that of the MMSE equalizer, and the larger the block length, the larger the accuracy.

Algorithm has been tested on mixtures of length \(K = 3\). We have generated 100 random channels. The equalizer has been determined for each channel with data blocks of 256 symbols or 512 symbols. Then, the equalizer has been tested with two different realizations of 5000 symbols each in order to compute the Symbol Error Rate (SER). Thus, the
minimal resolution is \((2 \times 5000 \times 100)^{-1} = 10^{-6}\).

Figure 3 shows median results of the 100 trials for QPSK signals, i.e., median is an estimate of average SER. This figure proves that algorithm works well on short data length since from only 256 symbols and with a noise of 16dB, the median SER is below the minimal resolution.

For each trial, the MMSE solution is computed. Another basis for comparison is the distance (Frobenius norm) between the equalizer found and the Zero-Forcing (ZF) solution. In fact, the ZF solution for MIMO convolutive mixtures is simply the inverse of the channel. Then, when \(K = L\), the global transfer function with the ZF equalizer \(H_{ZF}[z]\) is

\[
G[z] = \sum_n \sum_m C(m)H_{ZF}(n)z^{-(n-m)} = I z^{-L}.
\]

(28)

So, we just have to compare the global transfer function \(G[z]\) obtained with the paraunitary equalizer, to the identity matrix up to a multiplicative matrix \(D[z]\) of the form \(\Lambda[z]P\), where \(\Lambda[z]\) is diagonal and \(P\) a permutation. In order to do this, we proceed by determining the best matrix \(D[z]\) as follows. Once \(G[z]\) is estimated, we store in an \(N \times N(2L - 1)\) array \(\mathcal{G}\) by merely stacking the matrices one after the other. Next, we search for columns containing the entry of largest modulus in each row of \(\mathcal{G}\). By comparing column indices, which have to be different modulo \(N\), we find matrix \(P\) and \(\Lambda[z]\). Next, we normalize each row by its entry of largest modulus. The matrix obtained this way is denoted \(\mathcal{G}\). Then, we replace the \(N\) previous entries of \(\mathcal{G}\) by zeros. This is the same as subtracting \(D\) to \(\mathcal{G}\), where \(D\) is made from \(D[z]\) with the same size as \(\mathcal{G}\). Finally, we compute the Frobenius norm of the resulting matrix, i.e., \(\|\mathcal{G} - D\|\). This value is the distance between the paraunitary equalizer found and the ZF solution. Medians of distances are reported in figure 4. We note that between 0 and 8dB and for 512 symbols, the distance decreases rapidly from 1 to 0.25. Then, after 8dB, the distance remains approximately constant.

7. Concluding remarks

The Blind MIMO Equalization problem can be reduced to a simpler one via a prior space-time second-order whitening, namely the equalization of a paraunitary channel. Thanks to this statement, we have proposed a parameterization of paraunitary equalizers, in order to carry out blind source separation, i.e., deconvolution without a learning sequence. Based on this parameterization, a numerical algorithm has been devised, which iteratively maximizes a contrast through successive sweeps of plane rotations, and eventually equalizes the channel. Moreover, we have demonstrated that the computation of the blind equalizer can be completed within a polynomial complexity, useful in burst-mode transmissions. Performances presented in the last section report an average SER of 0.35% at 12dB of SNR, with data length of 256 symbols only. Other issues currently being addressed include robustness to chan-
nel length misadjustment, which are of prime importance as pointed out in [32,34], and space-time whitening implementations.

Appendix A. Complexity savings

In order to reduce the complexity of the algorithm, it is suggested to store each $K$ in a matrix $T \in \mathbb{C}^{N^4 \times N^4}$. This matrix is defined by

$$T \equiv \sum_{i,j} f_i(\tau)c(\tau)^T. \quad (A.1)$$

where, for $N = 2$,

$$f_i(\tau) = \begin{bmatrix}
    f_i^{111}(\tau_1, \tau_2, \tau_3, \tau_4) \\
    f_i^{112}(\tau_1, \tau_2, \tau_3, \tau_4) \\
    \vdots \\
    f_i^{2222}(\tau_1, \tau_2, \tau_3, \tau_4)
\end{bmatrix}$$

and

$$c(\tau) = \begin{bmatrix}
    \Gamma_1^{11}(\tau_1, \tau_2, \tau_3, \tau_4) \\
    \Gamma_1^{12}(\tau_1, \tau_2, \tau_3, \tau_4) \\
    \vdots \\
    \Gamma_2^{22}(\tau_1, \tau_2, \tau_3, \tau_4)
\end{bmatrix}. \quad (A.3)$$

For only 2 observations, we have $T = (N + 1)^4$ loops for building vectors $f_i$ and $c$, e.g. by Kronecker product. In this case, both vectors are of length $(N(L + 1))^4$ and then $T$ is $(N(L + 1))^4 \times (N(L + 1))^4$. So, we only have $(N(L + 1))^4$ loops for building vectors $f_i$ and $c$, and then we have to make only one product $f_i^Tc$ for matrix $T$. Nevertheless, the memory capacity is then the drawback of this method since more memory is used for storing matrix $T$. As a consequence, this last simplification is probably interesting for small values of $L$.

References


