LINK BETWEEN THE JOINT DIAGONALISATION OF SYMMETRICAL CUBES AND PARAFAC: AN APPLICATION TO SECONDARY SURVEILLANCE RADAR

Nicolas PETROCHILOS

CReSTIC, University of Reims
Moulins de la Housse, BP 1039
51687 Reims Cedex 2, France
email: petro@ieee.org

Pierre COMON

I3S–CNRS, University of Nice, Algorithmes,
2000 route des Lucioles, BP 121,
06903 Sophia-Antipolis, France
email: pcomon@i3s.unice.fr

ABSTRACT

The Manchester Decoding Algorithm (MDA) presented in [1] succeeds in separating Secondary Surveillance Radar (SSR) replies impinging on an array. The final step of the MDA consist of jointly diagonalizing a collection of several symmetric cubes by a sub-optimal technique. In this article, we demonstrate that it is in fact a PARAFAC problem with an almost symmetric solution. Furthermore, comparisons with other algorithms are carried out, with the help of computer simulations.

1. INTRODUCTION

Secondary Surveillance Radar (SSR) is essential for Air Traffic Control (ATC). This radar establishes a communication-link between the ground-station and the aircraft, in which the aircraft identity or altitude is transmitted on a request and reply basis. Under the present circumstances of operation, SSR uses two protocols, while in the future, only mode S will be operated [2].
It has been foreseen to replace the conventional rotating antenna by a distribution of receiving passive array antenna. The expected gain from this change is a better localization of the airplane based on multi-lateration principle (as in e.g. [3]). The drawback is that more replies are received due to the antennas omni-directionality. On the other hand, using an array allows to perform source separation on incoming replies.
With this goal, several algorithms tailored to this application have been proposed [4, 5, 6, 1]. These Algorithms need in their final step to solve either a joint diagonalization of matrices, or a Generalized Eigenvalue Problem, or even a Quadratic Eigenvalue Problem. Let d be the number of sources. One of the algorithms, the Manchester Decoding Algorithm (MDA), needs to jointly diagonalize a collection of d third order symmetric tensors of dimension $d \times d \times d$. We show in this article that it is equivalent to find the PARAFAC decomposition of a fourth-order tensor.

2. MODEL

At base-band, a received mode S SSR reply consists of a binary sequence with alphabet $\{0, 1\}$, modulated by a complex exponential due to a residual carrier frequency [2]. Moreover the binary sequence is encoded in a Manchester scheme: a bit $b_n = 0$ is coded as $\mathbf{b}_n = [0, 1]$, and a bit $b_n = 1$ as $\mathbf{b}_n = [1, 0]$. Therefore three consecutive samples of a reply have a product equal to zero (see Figure 1):

Property 1 *Independently of the data transmitted, due the Manchester Encoding, a mode S reply signal $s[n]$ obeys:*

$$s[n-1]\; s[n]\; s[n+1] = 0\,, \qquad \forall n \in \mathbb{N} \tag{1}$$

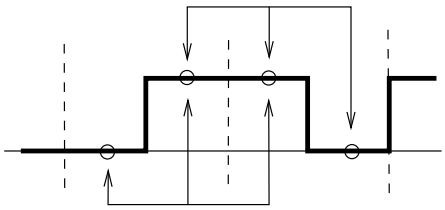

Fig. 1. The cross-product of three consecutive bits is always equal to zero, with a Manchester encoding.

It is assumed that a M-element antenna receives d replies. After sampling at twice the data rate, the received signal can be modeled as an instantaneous linear combination of source signals:

$$\mathbf{x}[n] = \mathbf{M}\mathbf{s}[n] + \mathbf{n}[n] \quad \forall n \in \{1, .., N\} \tag{2}$$

where

- N is the total number of samples,
- $\mathbf{x}[n] = [x_1[n], \cdots, x_p[n]]^T$ is a $p \times 1$ vector of received signals at time nT,

- $\mathbf{M}$ is the $M \times d$ mixing matrix, taking account the direction of arrival, and a possible decalibration of the array,
- $\mathbf{s}[n] = [s_1[n], \cdots, s_d[n]]^T$ is a $d \times 1$ vector of source signals sampled at time n, $s_i[n]$ being the n-th sample of the i-th source,
- $\mathbf{n}[n]$ is a white Gaussian noise.

Note that the mixture is instantaneous because possible multipaths affect only $\mathbf{M}$, or have so large time delays that they may be considered as an independent sources.

3. ALGORITHMS

3.1. Manchester Decoding Algorithm

The aim of [1] is to find beamformers $\mathbf{w}_i$, $\forall i \in \{1, .., d\}$, such that the outputs:

$$\mathbf{w}_i^H \mathbf{x}[n] = \hat{s}_i[n]$$

are the SSR replies; this is a Zero-Forcing solution (ZF). Using property (1), and replacing $s_i[n]$ by $\mathbf{w}^H \mathbf{x}[n]$, one obtains:

$$[\mathbf{x}[n+1] \otimes \mathbf{x}[n] \otimes \mathbf{x}[n-1]]^H (\mathbf{w} \otimes \mathbf{w} \otimes \mathbf{w}) = 0 \quad (3)$$

for $n = 2, \cdots, N-2$, where $\otimes$ is the Kronecker product. To collect these conditions, define the matrix $\mathbf{P} : N-2 \times d^3$ built by stacking rows $[\mathbf{x}[n+1] \otimes \mathbf{x}[n] \otimes \mathbf{x}[n-1]]^H$, so that

$$\mathbf{P}\mathbf{w}^\diamond = 0\,, \quad \text{with } \mathbf{w}^\diamond \stackrel{\text{def}}{=} \mathbf{w} \otimes \mathbf{w} \otimes \mathbf{w} \quad (4)$$

For d sources, there are d linearly independent separating beamformers $\mathbf{w}_i$, $i = 1, \cdots, d$. Thus we have d linearly independent vectors $\mathbf{w}_i^\diamond$ that belong to the null space of $\mathbf{P}$. If the null space is d-dimensional, then the subspace spanned by the $\mathbf{w}_i^\diamond$ is exactly equal to the null space, and any basis of the null space must be a linear combination of the $\mathbf{w}_i^\diamond$'s.

In [5], a proposition states that the null space of $\mathbf{P}$ is d-dimensional if the replies are totally overlapping. The MDA, similarly to [7], estimates an arbitrary basis of the null space of matrix $\mathbf{P}$; a matrix $\mathbf{U}$ such that:

$$\mathbf{U} = [\mathbf{w}_1^\diamond \mathbf{w}_2^\diamond \cdots \mathbf{w}_d^\diamond]\, \mathbf{T} = \mathbf{W}^\diamond \mathbf{T} \quad (5)$$

where $\mathbf{T}$ is an invertible matrix. Let $\mathbf{u}_i$ be the i-th column of $\mathbf{U}_0$ with size d^3, then $\mathbf{u}_i = \sum_{j=1}^d (\mathbf{T})_{ij}\, \mathbf{w}_j^\diamond$.
The last step of the algorithm is to find the linear combinations to map the basis to the structured vectors $\mathbf{w}_i^\diamond$, and subsequently to estimate the corresponding $\mathbf{w}_i$ for each vector.

From each vector $\mathbf{u}_i$ (of size d^3) of the basis $\mathbf{U}$, it is possible to create a tensor $\mathbb{U}_i$ of size $d \times d \times d$, a "cube" of data made by rearranging entries of $\mathbf{u}_i$ in $\mathbb{U}_i$ along the first dimension, then the second, and lastly the third. Then:

$$\mathbb{U}_i = \sum_{l=1}^d \mathbf{T}_{l,i} \mathbb{W}_l \quad (6)$$

where tensors $\mathbb{W}_i$'s are produced by the $\mathbf{w}_i^\diamond$'s. These d symmetric tensors $\mathbb{U}_i$ can be jointly diagonalized by the same transform $\mathbf{Q}$, which allows to determine the desired d beamformers.

In [1], the joint diagonalization is indirect, indeed the columns $\mathbf{u}_i$'s are reshaped into $d^2 \times d$ matrices $\mathbf{U}_i$, such that $\text{vec}(\mathbf{U}_i) = \mathbf{u}_i$. Then

$$\mathbf{U}_i = \sum_{j=1}^d (\mathbf{T})_{ij}\, (\mathbf{w}_j \otimes \mathbf{w}_j)\mathbf{w}_j^T = (\mathbf{W} \circ \mathbf{W})\mathbf{\Lambda}_i \mathbf{W}^T$$

where $\mathbf{\Lambda}_i$ is a diagonal matrix which entries are $(\mathbf{\Lambda}_i)_{jj} = (\mathbf{T})_{ij}$. We can define the $d^2 \times d^2$ matrix $\mathbf{U}'$ as

$$\begin{aligned} \mathbf{U}' &\stackrel{\text{def}}{=} [\mathbf{U}_1, \cdots, \mathbf{U}_d] \\ &= (\mathbf{W} \circ \mathbf{W})[\mathbf{\Lambda}_1 \mathbf{W}^T, \cdots, \mathbf{\Lambda}_d \mathbf{W}^T] \end{aligned} \quad (7)$$

This shows that $\mathbf{U}'$ should be of rank d. Thus let $\mathbf{V}$ be an estimated d-dimensional basis for the column span of $\mathbf{U}'$, obtained via an SVD of $\mathbf{U}'$, and let $\mathbf{Q} = \mathbf{V}^H(\mathbf{W} \circ \mathbf{W})$, with size $d \times d$. Then $\mathbf{V}^H \mathbf{U}'$ is a dimension-reduced ($d \times d^2$) matrix with square $d \times d$ blocks $\mathbf{V}^H \mathbf{U}_i$, each of the form

$$(\mathbf{V}^H \mathbf{U}_i) = \mathbf{Q}\mathbf{\Lambda}_i \mathbf{W}^T\,, \qquad i = \{1, \cdots, d\}$$

Thus, the problem is reduced to a standard (unsymmetric) joint diagonalization problem, and the algorithm in [7] can be applied to estimate $\mathbf{W}$.

3.2. PARAFAC model

A PARAFAC decomposition of a third-order tensor of rank r can be described as the sum of r rank-one tensors:

$$\mathbb{T}_{abc} = \sum_{l=1}^r \mathbf{A}_{al} \mathbf{B}_{bl} \mathbf{C}_{cl}$$

where $\mathbf{A} = [\mathbf{a}_1 \cdots \mathbf{a}_r]$, $\mathbf{B} = [\mathbf{b}_1 \cdots \mathbf{b}_r]$, and $\mathbf{C} = [\mathbf{c}_1 \cdots \mathbf{c}_r]$ are matrices of dimension $k_i \times r$. It can be then written as:

$$\mathbb{T} = \sum_{l=1}^r \mathbf{a}_l \circ \mathbf{b}_l \circ \mathbf{c}_l = \mathbf{A} \circ \mathbf{B} \circ \mathbf{C}$$

where $\circ$ denotes the outer product between two tensors (see [8]).
Similarly, define $\mathbf{D} = [\mathbf{d}_1 \cdots \mathbf{d}_r]$ a matrix of dimension

$k_4 \times r$, then the PARAFAC decomposition of a fourth-order tensor of rank r can be written:

$$\mathbb{T} = \sum_{l=1}^{r} \mathbf{a}_l \circ \mathbf{b}_l \circ \mathbf{c}_l \circ \mathbf{d}_l = \mathbf{A} \circ \mathbf{B} \circ \mathbf{C} \circ \mathbf{D}$$

where the entry in position $\{abcd\}$ is:

$$\mathbb{T}_{abcd} = \sum_{l=1}^{r} \mathbf{A}_{al} \mathbf{B}_{bl} \mathbf{C}_{cl} \mathbf{D}_{dl}$$

3.3. The MDA: a PARAFAC problem?

First of all, note that the $\mathbb{W}_l$ are rank one since $\mathbb{W}_l = \mathbf{w}_l \circ \mathbf{w}_l \circ \mathbf{w}_l$. Therefore equation (6) may be written as:

$$\mathbb{U}_i = \sum_{l=1}^{d} \mathbf{T}_{l,i}\, \mathbf{w}_l \circ \mathbf{w}_l \circ \mathbf{w}_l$$

where $\mathbb{U}_i$ is a square symmetric tensor of order 3, dimension d, and rank d. Next, let us build the fourth-order tensor $\mathbb{U}^{(4)}$ such that the i-th slice along the last dimension is $\mathbb{U}_i$: $\mathbb{U}^{(4)}(:,:,:,i) = \mathbb{U}_i$.
Then, this tensor is equal to:

$$\mathbb{U}^{(4)} = \sum_{l=1}^{r} \mathbf{w}_l \circ \mathbf{w}_l \circ \mathbf{w}_l \circ \mathbf{t}_l = \mathbf{W} \circ \mathbf{W} \circ \mathbf{W} \circ \mathbf{T}$$

and is of rank d, therefore the PARAFAC model applies to the tensor issued for the final resolution of the MDA problem.

3.4. Pre-processing

The $\mathbb{U}_i$ are a sum of square symmetric tensors, therefore they should be symmetric as well, but due to noise, and to the method used to estimate them, it is often not the case. Given that the desired solutions are symmetric in ways $(1,2,3)$, we pre-process the received data in order to symmetrize the $\mathbb{U}_i$. Denote $(.)_{[abc]}$ the permutation of indices in the first 3 ways, where ways are re-arranged in the order $[abc]$; the symmetrized version of the $\mathbb{U}_i$'s takes the form:

$$\begin{aligned}\hat{\mathbb{U}}_i &= \frac{1}{6}\left((\mathbb{U}_i)_{[123]} + (\mathbb{U}_i)_{[231]} + (\mathbb{U}_i)_{[312]}\right) \\ &\quad + \frac{1}{6}\left((\mathbb{U}_i)_{[213]} + (\mathbb{U}_i)_{[321]} + (\mathbb{U}_i)_{[132]}\right)\end{aligned}$$

During the simulations, we try both cases.

3.5. Rank-one approximation

In the process of the various algorithms adapted or proposed in this article, we need a tool that can give the rank-one approximation of a square symmetric third-order tensor $\mathbb{T}$ of rank ideally equal to 1:

$$\mathbb{T} \approx \mathbf{u} \circ \mathbf{u} \circ \mathbf{u} \tag{8}$$

We propose two approaches: one is derived from the Power Method, and the other is an ad-hoc technique based on a Singular Value Decomposition.
Remind that $(\bullet_i)$ is the inner product between two tensors along way i (see [8]). Then Equation (8) gives:

$$\mathbb{T} \underset{2}{\bullet} \mathbf{u}^* \underset{3}{\bullet} \mathbf{u}^* \approx \mathbf{u}$$

The *pseudo*-Power Method consists to cyclically multiply the tensor $\mathbb{T}$ along two dimensions, and to use the result as a new estimate for the next multiplication.
Rk: This technique should be applied only to model (8), and not with general tensors[1].

In the second method, the idea is to use all the slices (in any dimension) of the cube. Since the tensor is square, the number of slices is $3d$. Denotes $\mathbb{T}_{(:,i,:)}$, the i-th slice along the second dimension, it is equal to: $\mathbb{T}_{(:,i,:)} = \mathbf{u}_i \mathbf{u}\ \mathbf{u}^T$, which is symmetric. We construct the matrix of size $d \times 6d^2$:

$$\mathbf{M} = \left[\mathbb{T}_{(:,:,1)}\mathbb{T}^T_{(:,:,1)}\mathbb{T}_{(:,:,2)}\mathbb{T}^T_{(:,:,2)} \cdots \mathbb{T}_{(:,1,:)} \cdots \mathbb{T}^T_{(d,:,:)}\right].$$

The estimate of $\mathbf{u}$ is the main left singular eigenvector.

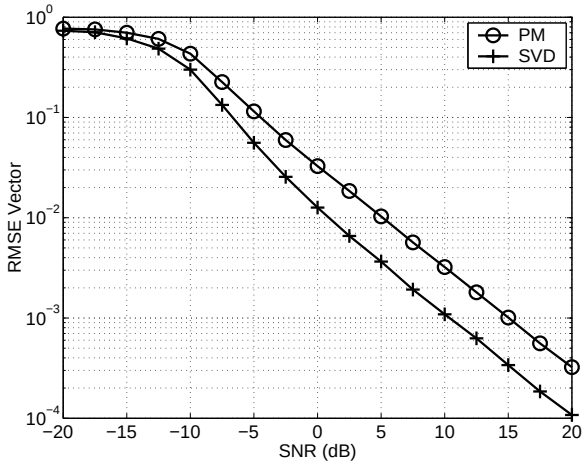

Fig. 2. RMSE of the estimated vector as a function of SNR, for the pseudo-**PM**, and the ad-hoc **SVD** technique.

In order to assess the quality of each method, we simulate a rank one square symmetric third-order tensor of dimension $5 \times 5 \times 5$ plus a noise tensor, whose entries are zero-mean Gaussian. The SNR is defined as the ratio of the Frobenius norm of the signal tensor over the noise tensor.

[1] A striking counter-example would be to try this procedure on a tensor such as: $\mathbb{T} = (\mathbb{U}_i)_{[123]} + (\mathbb{U}_i)_{[231]} + (\mathbb{U}_i)_{[312]}$, where $\mathbb{U} = \mathbf{u}_1 \circ \mathbf{u}_2 \circ \mathbf{u}_2 + \mathbf{u}_2 \circ \mathbf{u}_1 \circ \mathbf{u}_1$ with an initial $\mathbf{u} = \mathbf{u}_i$, and $\mathbf{u}_1 \perp \mathbf{u}_2$.

The RMSE vector is the power of the noise orthogonal to the direction of $\mathbf{u}$. The simulation is run over 1000 independent runs. Figure 2 demonstrates that the ad-Hoc method is more efficient.

3.6. Competiting Algorithms

• One historical algorithm to find the PARAFAC decomposition is the Alternating Least Square algorithm (ALS) [9] [10], which estimates Alternatively in the Least Square sense the matrices $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$ of the decomposition until convergence. Unfortunately, the ALS algorithm delivers the matrices $\{\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3, \mathbf{T}\}$, where the $\mathbf{W}_i$ would converge towards the same matrix $\mathbf{W}$ only in the absence of (unsymmetric) noise. Therefore, a final averaging between the $\mathbf{W}_i$'s is necessary.
An alternate solution, which we call Modified ALS (M-ALS), tends to overcome this drawback. Note that if $\mathbf{A} = \mathbf{B} = \mathbf{C} = \mathbf{W}$, we may only alternate between $\mathbf{W}$ and $\mathbf{D}$. Let $\mathbf{X}_D^T = \left[\mathbb{T}_{(1,1,:,:)}^T \mathbb{T}_{(1,2,:,:)}^T \cdots \mathbb{T}_{(2,1,:,:)}^T \cdots \mathbb{T}_{(d,d,:,:)}^T\right]$, then:

$$\mathbf{X}_d = (\mathbf{W} \odot \mathbf{W} \odot \mathbf{W})\, \mathbf{D}^T,$$

where $\odot$ is the Khatri-Rao product. So at the iteration $k+1$, we have:

$$\begin{cases} \mathbf{D}^{(k+1)} = \left[\left(\mathbf{W}^{(k)} \odot \mathbf{W}^{(k)} \odot \mathbf{W}^{(k)}\right)^{\dagger} \mathbf{X}_d\right], \text{ and} \\ (\mathbf{W} \odot \mathbf{W} \odot \mathbf{W})^{(k+1)} = \mathbf{X}_d \left((\mathbf{D}^{(k+1)})^T\right)^{\dagger} \end{cases}$$

where we denote $\mathbf{W}^{\diamond} = (\mathbf{W} \odot \mathbf{W} \odot \mathbf{W})^{(k+1)}$. Now, we need to estimate the $\mathbf{w}_i$'s: each vector of $\mathbf{W}^{\diamond}$, of size d^3, is transformed into a square cubic tensor, and is equal to $\mathbf{w}_i \circ \mathbf{w}_i \circ \mathbf{w}_i$. We estimate the $\mathbf{w}_i$'s by a rank-one approximation of the $\mathbb{W}_i$'s.

• The ACDC algorithm [11] consisting of two phases, AC and DC, jointly diagonalize a collection of matrices $\mathbf{A}_k$'s with the model $\mathbf{A}_k = \mathbf{B}\Lambda_k\mathbf{B}^{\star}$, where $(.)^{\star}$ can be either transpose $(.)^T$ or transpose-conjugate $(.)^H$. It is also a modified ALS, indeed it iterates along dimensions $(1, 2)$, the AC step, and (3), the DC step. Moreover, the DC step is identical to the corresponding ALS step. The difference lies on the estimation of $\mathbf{B}$, which is equal to $\mathbf{A}$ for the ACDC. Indeed each column $\mathbf{b}_i$ is obtained by the minimization of the cost function:

$$C_{LS} = \sum_{k=1}^{K} w_k \left\|\mathbf{A}_k - \mathbf{B}\lambda_k\mathbf{B}^T\right\|_{FRO}^2$$

while keeping the other columns $\mathbf{b}_j$, $j \neq i$, and the λ_k's constant. The w_k are some positive weights.
We extended the ACDC algorithm to a collection of symmetric tensors, in the simulation it is named **ACDC-3**. The algorithm is the same, except for the estimation of each $\mathbf{b}_i$, which is the result of the minimization of the cost function:

$$C_{LS}(\mathbf{b}_i) = \sum_{k=1}^{K} w_k \left\|\mathbb{U}_k - \sum_{l=1}^{d} \lambda_l^{[k]} \mathbf{b}_l \circ \mathbf{b}_l \circ \mathbf{b}_l\right\|_{FRO}^2$$

Defining $\mathbf{b}_i = r\boldsymbol{\beta}$, with $\boldsymbol{\beta}$ an unit-norm vector, and r a real valued-scalar. Be $\mathbb{U}'_k = \mathbb{U}'_k - \sum_{l\neq i}^{d} \lambda_l^{[k]} \mathbf{b}_l \circ \mathbf{b}_l \circ \mathbf{b}_l$, and $\mathbb{P} = \sum_{k=1}^{K} w_k \lambda_l^{[k]} (\mathbb{U}'_k)^*$, and $p = \sum_{k=1}^{d} w_k |\lambda_l^{[k]}|^2$, then:

$$\begin{cases} \boldsymbol{\beta} = \text{Argmax Re}\{\mathbb{P} \bullet_1 \boldsymbol{\beta} \bullet_2 \boldsymbol{\beta} \bullet_3 \boldsymbol{\beta}\} \\ r^3 = \frac{1}{p}\text{Re}\{\mathbb{P} \bullet_1 \boldsymbol{\beta} \bullet_2 \boldsymbol{\beta} \bullet_3 \boldsymbol{\beta}\}. \end{cases}$$

• JADE [12] jointly diagonalize a collection of matrices $\mathbf{A}_k$'s, with the model $\mathbf{A}_k = \mathbf{U}\Lambda_k\mathbf{U}^H$, where $\mathbf{U}$ is unitary. As well, the algorithm STOD from [13] jointly diagonalize a collection of third-order tensors by an unitary transformation, like JADE. Because the desired $\mathbf{W}$ is not unitary, these two techniques are not applicable in our case.

• Another approach is to minimize directly the criterion:

$$\mathcal{C}(\mathbf{W}) = \left\|\mathbb{U}^{(4)} - \mathbf{W} \circ \mathbf{W} \circ \mathbf{W} \circ \mathbf{T}\right\|_{\text{FRO}}^2 \tag{9}$$

With the notation from subsection 3.1, define $\Pi_{\mathbf{W}}^{\perp} = \mathbf{I} - \mathbf{W}^{\diamond}(\mathbf{W}^{\diamond})^{\dagger}$, where $(.)^{\dagger}$ is the Moore-Penrose pseudo-inverse. Then using Equation (5), the cost function can be rewritten:

$$\mathcal{C}(\mathbf{W}) = \left\|\Pi_{\mathbf{W}}^{\perp}\mathbf{U}\right\|_{\text{FRO}}^2 \tag{10}$$

This is the cost function we minimize using the Matlab© function `fminsearch.m`, we test it with two different initial point: 1) the true solution, e.g. $\mathbf{W} = \mathbf{M}^{\dagger}$, so we have a benchmark[2] for comparison. 2) the initial $\mathbf{W}$ is obtained by the joint diagonalization of the first slice of the tensors $\mathbb{U}_1$ and $\mathbb{U}_1$, denoted $\mathbf{A}_1$ and $\mathbf{A}_2$, so two matrices. the procedure is to perform the eigendecomposition of $\mathbf{A}_1^{-1}\mathbf{A}_2$ which yields the (joint) eigenvectors.

4. SIMULATION

Every aforementioned algorithm has been implemented both with the raw tensor $\mathbb{U}^{(4)}$, and its symmetrized version, $\mathbb{U}_S^{(4)}$ (dashed lines in the figures). 3 sources impinge on a 3-element array from directions $[60, 90, 120]$. Matrix $\mathbf{M}$ has a conditioning number of 1.58, and the maximum antenna gain is 3.02 dB. The data-batch is 100 samples long, the SNR range is $[0,\ 20]$ dB, and 1000 independent runs are executed.

Figure 3 reports the failure rate as a function of SNR, and Figure 4 the output SINR minus the input SNR. To declare a failure, we use a 3 dB threshold as it corresponds to the radar concept of tangential sensitivity. First, we note that

[2] It gives the Zero-Forcing solution.

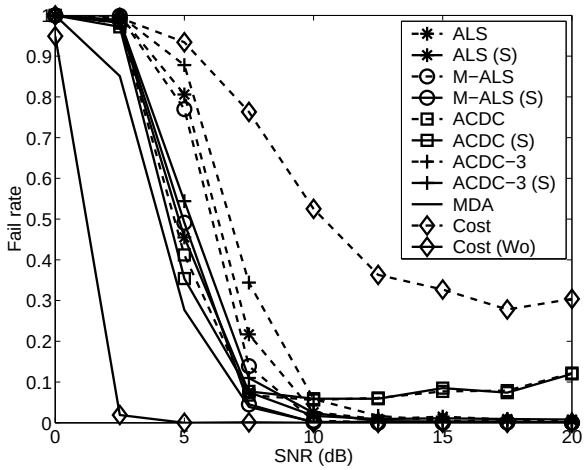

Fig. 3. Failure rate for each method as a function of SNR.

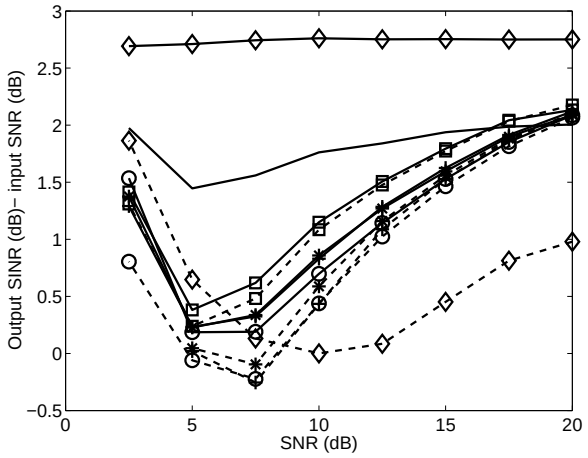

Fig. 4. The output SINR minus the input SNR for each method as a function of SNR.

the algorithms applied on $\mathbb{U}_S^{(4)}$ behave better for failure rate and SINR than when applied on $\mathbb{U}^{(4)}$. Next, the cost function minimization needs to start near the solution otherwise it gets lost in a local minimum. Initialized by the ZF solution, it has only 0.25 dB loss with respect to the maximum achievable SINR. The ALS, M-ALS, ACDC, and ACDC-3 behave similarly (within a 0.5 dB fork for SINR). The MDA performs better a SNR less than 17 dB. We think that the indirect step (7) implicitly performs a enhanced noise reduction, whose benefit overcomes the sub-optimality of the method; this effect is lost when the SNR is larger.

5. CONCLUSIONS

We presented how the final step of an algorithm that aims at separating a mixture of Secondary Surveillance Radar replies is linked to a PARAFAC model via an almost symmetric square fourth-order tensor of reduced rank. We have evaluated several algorithms performing a decomposition of this tensor. We have presented a technique based on SVD to perform a rank-one decomposition of a symmetric tensor. The ad-Hoc method from [1] outperforms at low SNR direct techniques, due to an implicit noise reduction. In future works, we shall investigate speed of convergence, identifiability, and enhanced noise reduction.

6. REFERENCES

[1] N. Petrochilos and A.J. van der Veen, "Algorithms to separe overlapping secondary surveillance radar replies," in *Proc. of ICASSP 2004*, 17-21 May 2004, pp. II.49–53.

[2] R.M. Trim, "Mode S: an introduction and overview," *Electronics & Communication Engineering Journal*, vol. 2, pp. 53–59, Apr. 1990.

[3] P. Bezousek, "A passive radar surveillance system VERA for ATC," in *IRS'98*, Munich, Germany, 1998.

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

[5] N. Petrochilos, "Algorithms for Separation of Secondary Surveillance Radar Replies," Phd thesis, Universiy of Nice-Sophia-Antipolis and TU Delft, Nice, France, July 2002, ISBN 90-407-2371-0, cas.et.tudelft.nl/∼nicolas, in english.

[6] S. Neugebauer, "Blind Channel Estimation and Equalization for Temporally-Correlated Constant-Modulus and Multi-Modulus Signals," Ph.d thesis, University of California, Davis, 2003.

[7] A.J. van der Veen and A. Paulraj, "An analytical constant modulus algorithm," *IEEE Trans. Signal Processing*, vol. 44, no. 5, pp. 1136–1155, May 1996.

[8] P. Comon, "Tensor decompositions, state of the art and applications," in *IMA Conf. Mathematics in Signal Processing*, Warwick, UK, Dec. 18-20, 2000, keynote address.

[9] R. A. Harshman, "Foundations of the Parafac procedure: Models and conditions for an explanatory multimodal factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, pp. 1–84, 1970.

[10] J. B. Kruskal, "Three-way arrays: Rank and uniqueness of trilinear decompositions," *Linear Algebra and Applications*, vol. 18, pp. 95–138, 1977.

[11] A. Yeredor, "Non-orthogonal joint diagonalization in the least-squares sensewith application in blind source separation," *IEEE Trans. on Signal Processing*, vol. 50, no. 7, pp. 1545–1553, July 2002.

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

[13] L. de Lathauwer, B. de Moor, and J. Vandewalle, "Independent component analysis and (simultaneous) third-ordertensor diagonalization," *IEEE Trans. on Signal Processing*, vol. 49, no. 10, pp. 2262–2271, Oct. 2001.

[14] A.J. van der Veen, "Joint diagonalization via subspace fitting techniques," in *Proc. IEEE ICASSP'01*, Salt Lake City (UT), May 2001.