AZIMUTH AND ELEVATION COMPUTATION IN HIGH RESOLUTION DOA ESTIMATION[1]

A.J. van der Veen, P.B. Ober and E.F. Deprettere

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

ABSTRACT

In this paper, we discuss a number of high-resolution direction finding methods for determining the two-dimensional directions of arrival of a number of plane waves, impinging on a sensor array. The array consists of triplets of sensors that are identical, as an extension of the 1D ESPRIT scenario to two dimensions. New algorithms are devised that yield the correct parameter pairs while avoiding an extensive search over the two separate one-dimensional parameter sets.

EDICS: 5.2.2., 5.2.4.

1. INTRODUCTION

Subspace based high resolution DOA algorithms considered in this paper (typically "ESPRIT"-like [1], see [2, 3, 4] for an overview), are usually designed to determine the directions of arrival of narrowband non-coherent signals in only one parameter dimension, i.e. array and waves are confined to a single plane. The extension to the 2D case, where both azimuth and elevation angles have to be determined, is in general non-trivial. The decomposition of the problem into two independent 1D problems results in two decoupled parameter sets, which have to be combined to correct parameter pairs. Subspace-based approaches to solve the 1D DOA problem typically end up with a matrix pencil, formed on the data after noise reduction by SVD-based algorithms. The generalized eigenvalues of the pencil contain (implicitly) the directions of arrival of the impinging signals. In the 1D based 2D case, two pencils are constructed on three data matrices, and, after having solved these pencils independently, the problem of eigenvalue matching occurs: an algorithm must be found to combine azimuths and elevations correctly. This parameter matching is essentially what makes the 2D problem more difficult to solve. Different approaches have been alluded to in the literature:

1. Correlation technique. This method is described in Zoltowski et al. [5]. After calculating the correlation matrix between the outputs of all sensors, compute the eigenvectors that correspond to the

[1] Research supported in part by the commission of the EC under the ESPRIT BRA program 3280 (NANA).

IEEE Trans. Signal Processing, 40(7):1828-1832, July 1992.

noise. The correct pairing is labeled as that for which the array manifold vector projection on the noise-subspace is 'closest to zero', and is followed by a MUSIC search in parameter space. This method is computationally not very attractive.

2. Overdetermined direction data. If the sensor array has spacial redundancy, a third matrix pencil can be constructed, having generalized eigenvalues equal to the quotient of the correct parameter pairs. These quotients can be used to search for the correct pairs. This method was also proposed in [5]. See section 3.1

Both these approaches are not particularly elegant, in the sense that they are not very well suited to combine with the VLSI integration of the ESPRIT algorithm as proposed in [3] for the 1D case. We would like to solve the problem with linear algebra, rather than with combinatorics.

In this paper, two new approaches are considered, both based on the observation that the data matrices share the same common set of eigenvectors in case no noise is present. Two matrices sharing the same eigenvectors can be diagonalized by the same similarity transform, and hence the two underlying pencils can be solved for by diagonalizing the first pencil, and applying the similarity transform that was needed to the second. The correct pairings are found directly. When noise is present, this property is lost. We will describe how this method can be adapted such that the correct eigenvalues of each pencil are determined, along with an estimate of the pairing. The first new approach is based on this idea, and yields good results, yet is "in style" with the VLSI parallel array architecture described in [3] and is extremely cheap in number of extra operations. See secion 3.2. The underlying property was already mentioned in [6] and was exploited there in some sense, along with spacial overdetermining to improve the resolution and correct pairing results.

The second new approach is based on the idea to approximate the data matrices by adding small perturbation matrices such that the resulting matrices will have equal eigenvectors, or, equivalently, that the two resulting matrices will commute. This approach was briefly reported in [7]. It was brought to our attention that a similar, although not identical, approach was presented independently by Swindlehurst and Kailath [8], who perturb only the second matrix such that its eigenvectors will coincide with the eigenvectors of the first (unperturbed) matrix.

In section 2, a short review of the ESPRIT data model is presented, along with a review of the corresponding matrix pencil approach and of the properties of the data in the noise free case. Section 4 discusses a number of approaches to enforce these properties on noisy data, in context with a review of some other algorithms. We finish with some simulation results.

2. PRELIMINARIES

2.1. The data model

Consider m sensor triplets, each composed of three identical sensors with unknown gain and phase patterns, which may vary from triplet to triplet. For every triplet, the displacement vectors $\mathbf{d}_{xy}$ and $\mathbf{d}_{xz}$ between its components are required to be the same (and, for convenience, are assumed to be orthogonal). This way, assigning the three sensor of each triplet to each of the sensor-arrays X, Y, Z, respectively, three identical although displaced arrays are obtained. This is a direct extension of the 1D ESPRIT scenario to two dimensions (see also [9]). Impinging on every array are d narrowband non-coherent signals $s_k(t)$, having an unknown complex amplitude $\hat{s}_k$ and a known center frequency ω_0:

$$s_k(t) = \hat{s}_k e^{j\omega_0 t} \qquad k = 1, 2, \ldots, d$$

Assuming the noise at the receiver to be additive, stationary and of zero mean, the output signal of the i-th sensor of each array will be

$$\begin{aligned} x_i(t) &= \textstyle\sum_{k=1}^{d} a_{ik}\, s_k(t) + n_{xi}(t) \\ y_i(t) &= \textstyle\sum_{k=1}^{d} a_{ik}\, \phi_k\, s_k(t) + n_{yi}(t) \\ z_i(t) &= \textstyle\sum_{k=1}^{d} a_{ik}\, \theta_k\, s_k(t) + n_{zi}(t) \end{aligned}$$

where a_{ik} is the gain of sensor i for signal $s_k(t)$, n_{xi}, n_{yi}, n_{zi} represent noise signals, and ϕ_k and θ_k are complex scaling factors of unit length, representing the phase shift caused by the differences in path length along the direction of propagation of incident waves. These numbers depend directly upon this direction, and, when $\mathbf{u}_k$ is the direction of propagation of $s_k(t)$, they can be written as

$$\begin{aligned} \phi_k &= e^{-j\omega_0(\mathbf{d}_{xy}^{\mathrm{T}} \cdot \mathbf{u}_k)/c} \\ \theta_k &= e^{-j\omega_0(\mathbf{d}_{xz}^{\mathrm{T}} \cdot \mathbf{u}_k)/c} \end{aligned} \tag{1}$$

in which c is the signal propagation velocity. By collecting N snapshots from each sensor, three data matrices X, Y and Z can be formed, obeying

$$\begin{aligned} X &= AS + N_x \\ Y &= A\Phi S + N_y \\ Z &= A\Theta S + N_z \end{aligned} \tag{2}$$

where $X(i, j) = x_i(t_j)$, $A(i,k) = a_{ik}$, $S(k, j) = s_k(t_j)$ and $N_x(i, j) = n_{xi}(t_j)$, and likewise for Y and Z. X, Y and Z are the data matrices (dimension $m \times N$), A is the array gain matrix ($m \times d$) and S is the signal matrix ($d \times N$). The matrices A and S are unknown, and are not rank-deficient by assumption. The matrices Φ and Θ are diagonal and contain the phase shifts (1) for each signal:

$$\begin{aligned} \Phi &= \mathrm{diag}(\phi_1, \phi_2, ..., \phi_d) \\ \Theta &= \mathrm{diag}(\theta_1, \theta_2, ..., \theta_d) \end{aligned}$$

The DOA problem is to estimate Φ and Θ. From these matrices, the angles of arrival can directly be computed.

2.2. The subspace approach

Matrix polynomials of the form $E - \alpha F, \alpha \in \mathbb{C}$, are called matrix pencils. Forming the pencils

$$\begin{aligned} X - \lambda Y &= A(I - \lambda\Phi)S + (N_x - \lambda N_y) \\ X - \mu Z &= A(I - \mu\Theta)S + (N_x - \mu N_z) \end{aligned}$$

it is seen that, in the noise free case, numbers $\lambda = \lambda_i$ and $\mu = \mu_j$, $i, j = 1, 2, \ldots, d$, that reduce the rank of the pencil by one are equal to ϕ_i^{-1} and θ_j^{-1} respectively. With square data matrices, these rank reducing numbers are the generalized eigenvalues of the matrix pairs (X,Y) and (X,Z).

With noise present, however, a large number of samples are taken to improve accuracy. As a result, X, Y and Z will not be square. Noise will also increase the rank of the pencils, and this will introduce new rank reducing numbers. By computing a Total Least Squares projection of the data matrices (see *e.g.,* [10] on this), and using the observation that, without noise, these matrices have a common (d-dimensional) row space and a common column space, it is possible to find square $d \times d$ matrices E_x, E_y and E_z, whose generalized eigenvalues are estimates of the d original rank reducing numbers. The next discussion closely follows the approach outlined in [3, 4]. Compute the SVD's of two matrices, constructed from X, Y and Z:

$$
\begin{aligned}
&[X \quad Y \quad Z] = U_1 \Sigma_1 V_1^{\mathrm{H}} \\
&\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = U_2 \Sigma_2 V_2^{\mathrm{H}} = \begin{bmatrix} U_{21} \\ U_{22} \\ U_{23} \end{bmatrix} \Sigma_2 V_2^{\mathrm{H}}
\end{aligned}
$$

where U_i and V_i are unitary matrices, Σ_i is a diagonal matrix containing the singular values, and '$^{\mathrm{H}}$' denotes herrmitian conjugation. Optimal (in the Frobenius norm) rank d approximations of these matrices are obtained by setting the $(m-d)$ smallest singular values in both Σ_1 and Σ_2 equal to zero, yielding $\hat{\Sigma}_1$ and $\hat{\Sigma}_2$ and approximations

$$
\begin{aligned}
&[X \quad Y \quad Z]\hat{} = \hat{U}_1 \hat{\Sigma}_1 \hat{V}_1^{\mathrm{H}} \\
&\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}\hat{} = \hat{U}_2 \hat{\Sigma}_2 \hat{V}_2^{\mathrm{H}}
\end{aligned}
\tag{3}
$$

in which $\hat{U}_i$, $\hat{V}_i$ contain the d columns of the U_i, V_i that correspond to the nonzero singular values in $\hat{\Sigma}_i$. The TLS approximations $\hat{X}$, $\hat{Y}$, $\hat{Z}$ of the data matrices share the same column space *range*$(\hat{U}_1)$ and row space *range*$(\hat{V}_2)$, as explicitly determined in (3), and are obtained by projecting X, Y and Z onto these subspaces:

$$
\begin{aligned}
\hat{X} &= (\hat{U}_1 \hat{U}_1^{\mathrm{H}})\, X\, (\hat{V}_2 \hat{V}_2^{\mathrm{H}}) &= \hat{U}_1 E_x \hat{V}_2^{\mathrm{H}} \\
\hat{Y} &= (\hat{U}_1 \hat{U}_1^{\mathrm{H}})\, Y\, (\hat{V}_2 \hat{V}_2^{\mathrm{H}}) &= \hat{U}_1 E_y \hat{V}_2^{\mathrm{H}} \\
\hat{Z} &= (\hat{U}_1 \hat{U}_1^{\mathrm{H}})\, Z\, (\hat{V}_2 \hat{V}_2^{\mathrm{H}}) &= \hat{U}_1 E_z \hat{V}_2^{\mathrm{H}}
\end{aligned}
$$

in which E_x, E_y and E_z are square $d \times d$ matrices, and with

$$
\begin{aligned}
E_x &= \hat{U}_1^{\mathrm{H}} X \hat{V}_2 = \hat{U}_1^{\mathrm{H}} \hat{U}_{21} \hat{\Sigma}_2 \\
E_y &= \hat{U}_1^{\mathrm{H}} Y \hat{V}_2 = \hat{U}_1^{\mathrm{H}} \hat{U}_{22} \hat{\Sigma}_2 \\
E_z &= \hat{U}_1^{\mathrm{H}} Z \hat{V}_2 = \hat{U}_1^{\mathrm{H}} \hat{U}_{23} \hat{\Sigma}_2
\end{aligned}
\tag{4}
$$

and

$$
\begin{bmatrix} \hat{U}_{21} \\ \hat{U}_{22} \\ \hat{U}_{23} \end{bmatrix} = \hat{U}_2
$$

The generalized eigenvalues $\mathrm{GE}(E_y, E_x)$, or the eigenvalues of $E_x^{-1} E_y$, are equal to the rank reducing numbers of the matrix pencil $\hat{X} - \lambda \hat{Y}$. The same holds true for the generalized eigenvalues of (E_z, E_x) and the pencil $\hat{X} - \lambda \hat{Z}$. To determine these eigenvalues from the E_i, multiplication by $\hat{\Sigma}_2$ can even be omitted, because this does not influence the result. For a more detailed discussion of the operations involved, and hints to a VLSI implementation, see [3].

Substituting (2) in (4) and defining $A_e = \hat{U}_1^{\mathrm{H}} A$ and $S_e = S\hat{V}_2$, we arrive at

$$\begin{array}{rcl} E_x &=& A_e S_e + N_{x,e} \\ E_y &=& A_e \Phi S_e + N_{y,e} \\ E_z &=& A_e \Theta S_e + N_{z,e} \end{array} \tag{5}$$

where $N_{x,e} = \hat{U}_1^{\mathrm{H}} N_x \hat{V}_2$, and similarly for $N_{y,e}$ and $N_{z,e}$.

2.3. Properties of E_x, E_y and E_z

When we define

$$\begin{array}{rcl} E_1 &=& E_x^{-1} E_y = S_e^{-1} \Phi S_e \\ E_2 &=& E_x^{-1} E_z = S_e^{-1} \Theta S_e \end{array} \tag{6}$$

it is clear that E_1 and E_2, whose eigenvalues we want to compute, share the same set of eigenvectors. This means they can be triangularized by the same unitary matrix Q: there exist unitary matrices Q_1, Q_2 such that

$$\begin{array}{rcccl} Q_1^{\mathrm{H}} E_1 Q_1 &=& Q_1^{\mathrm{H}} S_e^{-1} \Phi S_e Q_1 &=& R_1 \\ Q_2^{\mathrm{H}} E_2 Q_2 &=& Q_2^{\mathrm{H}} S_e^{-1} \Theta S_e Q_2 &=& R_2 \end{array} \tag{7}$$

with $Q_1 = Q_2 = Q$, and the upper triangular matrices R_1 and R_2 have main diagonals equal to Φ and Θ, respectively. Because the same matrix Q will triangularize both E_1 and E_2, the one-to-one correspondence between the ϕ_i's and θ_i's is preserved in the positional correspondence on the diagonals, and no pair matching operation needs to be done. Of course, due to noise, E_a will be of full rank, and Q_1 will differ somewhat from Q_2. This difference is assumed to be only small in the 'rotational perturbation' approach in section 3.2.

To assess the difference in eigenvectors of E_1 and E_2, recall that two matrices with the same eigenvectors commute. Thus, in the noise-free case,

$$E_1 E_2 = E_2 E_1$$

We will devise, also in the next section, an algorithm that determines small additive perturbations of E_1 and E_2 such that the above relation holds.

3. A BRIEF TOUR OF PAIR MATCHING ALGORITHMS

3.1. The searching algorithm

In the algorithm reported by Zoltowski et al. in [5], which we will name the 'searching algorithm', three instead of two pencils are formed:

$$\begin{array}{rcl} X - \lambda Y &=& A(I - \lambda\Phi)S \; + (N_x - \lambda N_y) \\ X - \mu Z &=& A(I - \mu\Theta)S \; + (N_x - \mu N_z) \\ Y - \nu Z &=& A(\Phi - \nu\Theta)S \; + (N_y - \nu N_z) \end{array}$$

From these equations we see that, without noise, the third pencil possesses rank reducing numbers ν that are exact quotients of the rank reducing numbers of the first two pencils:

$$\nu_i = \frac{\mu_i}{\lambda_i} = \frac{\phi_i}{\theta_i}$$

The searching method now tries to find two permutation matrices P_{b1} and P_{b2} that minimize the 'matching error'

$$\| \operatorname{diag}(R_1) - P_{b1}\operatorname{diag}(R_2)P_{b1}^{\mathrm{T}} \cdot P_{b2}\operatorname{diag}(R_3)P_{b2}^{\mathrm{T}} \|_F$$

where $\operatorname{diag}(R_1)$ contains the estimated ϕ_i, $\operatorname{diag}(R_2)$ contains the estimated θ_j and $\operatorname{diag}(R_3)$ the quotients ϕ_i/θ_i, determined from an extra $E_3 = E_z^{-1}E_y$. Note that the parameter values are estimated by solving the 1D pencils (X,Y) and (X,Z) independently, and the extra pencil is only used in estimating the correct pairing.

3.2. Matching using rotational perturbations

We now describe our first new approach to the matching problem. Experience gained with a Jacobi-iteration method for computing the Schur decomposition, as discussed in [3], led to the following pair matching algorithm, which can be integrated with the eigenvalue computations that are needed anyway. The Jacobi-iteration method consists of a number of sweeps, which in turn consist of a certain number of 2×2 elementary (Givens) rotations that solve 2×2 Schur decompositions. The elementary rotations are extended by a permutation scheme to ensure ultimate convergence. Two observations from [3] are that, near convergence, the Givens rotation angles are close to 0, so that the rotation matrix is close to the identity matrix, and that the permutation scheme is such that after an even number of sweeps the entries on the main diagonal have their initial ordering.

Triangularize E_1 as in (7) (in practise, do this with Generalized Eigenvalues): determine Q_1 to triangularize E_1, and simultaneously apply the same similarity transform to E_2:

$$\begin{array}{rcl} Q_1^{\mathrm{H}}E_1Q_1 & = & R_1 \\ Q_1^{\mathrm{H}}E_2Q_1 & = & R_2' \end{array} \qquad (8)$$

in which R_2' will be 'almost upper triangular' (in the noise free case, it would be completely upper triangular), and its diagonal entries are rough estimates of the true eigenvalues of E_2. Because of this, it is assessed that only 'small rotations' in the Jacobi iteration algorithm are needed to make R_2' upper triangular; i.e. there exists a unitary Q_2' that will triangularize R_2':

$$Q_2^{\mathrm{H}\prime}R_2'Q_2' = R_2$$

and which is close to the identity matrix in operator norm: Q_2' is a 'minimal rotational perturbation' of Q_1 such that $Q_2 = Q_1Q_2'$ triangularizes E_2. This means that Q_2' does not permute the rough eigenvalue estimates in R_2' while computing the correct eigenvalues of E_2. The correct eigenvalue pairs are thus the entries at the same position on the main diagonals of R_1 and R_2. Moreover, these eigenvalues are the eigenvalues of E_1 and E_2 and hence the same as obtained in e.g. Zoltowski's method: in effect, we have solved two 1D independent eigenvalue problems on (X,Y) and (X,Z), but in a special way that gives us the correct pairings almost 'for free'. (Remark that, while the eigenvalues are the same as in Zoltowski's method, the pairing may of course be different).

In addition, this algorithm is amenable to parallel VLSI implementation in the same way as the 1D algorithm was [3]: the only operations that are needed are elementary rotations (on E_1 and R_2'), and a multi-layer structure that allows the rotations that are performed on E_1 to be repeated on E_2 at the same time. Finally, we remark that 'permutation-less rotations' for Q_2' are obtained automatically when an *even* number of Jacobi-sweeps is performed in the iterative computation of the Schur decomposition of R_2': at the end of every second sweep, the ordering of the diagonal entries is their original ordering (see [3]).

3.3. Forcing commutativity of E_1 and E_2

In the presence of noise, E_1 and E_2 do not commute and can only be triangularized with different unitary matrices Q_1, Q_2:

$$\begin{array}{rcl} Q_1^{\mathrm{H}} E_1 Q_1 &=& R_1 \\ Q_2^{\mathrm{H}} E_2 Q_2 &=& R_2 \end{array}$$

where R_1 and R_2 are upper triangular, having diagonal entries that are approximations to the ϕ_i's and θ_i's of (1). It is not possible to find the correct tuples ϕ_k, θ_k directly, because their ordering along the diagonals of R_1 and R_2 may be different. Because the non-commutativity of E_1 and E_2 is caused by additive noise, the idea is to (partially) cancel this noise by adding perturbation matrices to them in such a way that their commutativity is restored. Thus, we are looking for two perturbation matrices P_1 and P_2 such that

$$\begin{array}{rl} & (E_1+P_1)(E_2+P_2) = (E_2+P_2)(E_1+P_1) \\ \Leftrightarrow & (E_1E_2-E_2E_1)+(P_1P_2-P_2P_1) = (E_2P_1-P_1E_2)-(E_1P_2-P_2E_1) \\ \Leftrightarrow & (E_1E_2-E_2E_1)+(P_1P_2-P_2P_1) = [E_2 \quad -E_1] \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} - [P_1 \quad P_2] \begin{bmatrix} E_2 \\ -E_1 \end{bmatrix} \end{array}$$

and such that P_1, P_2 obey some minimum norm constraint. Thus with *e.g.*, a Frobenius norm, the problem is

$$\begin{array}{ll} \min & \|P_1\|_F^2 + \|P_2\|_F^2 \\ \text{such that} & [P_1 \quad P_2 \quad I] \begin{bmatrix} -E_2 & 0 & I \\ E_1 & -I & 0 \\ E_2E_1-E_1E_2 & -E_2 & E_1 \end{bmatrix} \begin{bmatrix} I \\ -P_1 \\ -P_2 \end{bmatrix} = 0 \end{array} \qquad (9)$$

After determining the solution to this non-linear minimization problem, only the eigenvalue decomposition of (E_1+P_1) needs to be determined. The same unitary similarity transformation that will make (E_1+P_1) upper triangular will also triangularize (E_2+P_2). The assumedly more accurate parameter pairs follow directly from the entries of the diagonals of the resulting matrices.

The exact solution to this minimization problem is in general hard to find, but could be obtained by means of nonlinear programming. Note that equation (9) is a kind of generalized (constrained) Hamiltonian equation: it would represent a Ricatti equation if $[P_1 \ \ P_2]$ would be a square matrix S, and then the problem would be

$$[S \ \ I] \, \mathbf{H} \begin{bmatrix} I \\ -S \end{bmatrix} = 0$$

in which case S has a solution given in terms of the eigenvalues of $\mathbf{H}$. However, this insight does not really help in solving our problem. The non-square-ness makes our problem singular, and implies that there is a collection of perturbation matrices that makes the matrices E_1, E_2 commute.

An approximate solution for P_1 and P_2 that is easier to compute can however be obtained by neglecting the term $P_1P_2 - P_2P_1$, under the assumption that the perturbations are small in comparison with the E_i anyway. This results in a condition, linear in the entries of P_1 , P_2 that has close resemblance to a Lyapunov equation, viz.

$$BX - XB = C,$$

(with B, C square matrices however) which can be solved by Kronecker sums. Evaluating this last method, we arrive at the following equations:

$$\left[-E_2^{\mathrm{T}} \oplus E_2 \quad E_1^{\mathrm{T}} \oplus -E_1 \right] \cdot \begin{bmatrix} \mathrm{vec}(P_1) \\ \mathrm{vec}(P_2) \end{bmatrix} = \mathrm{vec}(E_1 E_2 - E_2 E_1)$$

(where the Kronecker sum $A \oplus B$ equals $A \otimes I + I \otimes B$, $\otimes$ is the Kronecker product, and vec(A) is a vector obtained by stacking the columns of A). From this equation we can find, using the Moore-Penrose pseudo-inverse denoted by '${}^{+}$', the solution with minimum perturbation norm,

$$\begin{bmatrix} \mathrm{vec}(P_1) \\ \mathrm{vec}(P_2) \end{bmatrix} = \left[-E_2^{\mathrm{T}} \oplus E_2 \quad E_1^{\mathrm{T}} \oplus -E_1 \right]^{+} \cdot \mathrm{vec}(E_1 E_2 - E_2 E_1) \tag{10}$$

While solving the above equation in this way is obviously a computational overkill compared with the original problem, it does find the minimum norm solution to the approximate problem, and can give hints to a (much more) efficient solution. It should be remarked that the usual algorithm to the Lyapunov problem involves an eigenvalue decomposition of B, and can not be used directly in our problem since the matrices are not square. Also note that, due to the approximation $P_1 P_2 - P_2 P_1 \approx 0$, the resulting matrices $(E_1 + P_1)$ and $(E_2 + P_2)$ do not commute precisely.

Another possible simplification of the problem (9) to make it solvable is to add a perturbation term only to E_2, *i.e.*, to take $P_1 = 0$ (see also Swindlehurst and Kailath [8, 11]). The resulting matrices E_1 and $E_2 + P_2$ do commute, and P_2 is obtained via a Kronecker sum:

$$\mathrm{vec}(P_2) = \left[E_1^{\mathrm{T}} \oplus -E_1 \right]^{+} \mathrm{vec}(E_1 E_2 - E_2 E_1) \tag{11}$$

Because E_1 is not perturbed, the estimation for Φ is the same as the estimation obtained in sections 3.1 and 3.2: $\Phi = \mathrm{eig}(E_1)$. The estimate for Θ, *i.e.*, the eigenvalues of $E_2 + P_2$, is, of course, in general different from the estimate obtained in the previous sections.

In this respect, note that the method of making matrices commute by adding small perturbation matrices to them is not too well-posed: if E_1 and E_2 are similarity-transformed to $E_1' = T E_1 T^{-1}$ and $E_2' = T E_2 T^{-1}$, then their eigenvalues stay the same, yet the minimum norm perturbations that must be added to make them commute are different (not equal to $T P_1 T^{-1}$ and $T P_2 T^{-1}$), and hence the resulting eigenvalues of $E_1' + P_1'$ and $E_2' + P_2'$ are also different: the solution is dependent on the initial parametrization of the problem. As an example, take T such that E_1' is diagonal (and equal to the estimated value of Φ), then E_2' is not yet diagonal, but can be made diagonal by adding a perturbation P_2'. It turns out that the minimum norm perturbation is such that the off-diagonal entries of E_2' are zeroed, while the main diagonal entries are left intact: the estimated value of Θ is $\Theta = \mathrm{diag}(E_2')$, which is not unlike the intermediate results of the 'rough estimate' preprocessing in section 3.2 (equation (8)).

4. SIMULATION RESULTS

To give an indication of the behaviour of the methods discussed in the previous chapter, we devised the following test scenario. In all simulations, the number of sources is $d = 4$, and their angles of incidence are

$(10^o, 25^o)$, $(15^o, 20^o)$, $(20^o, 15^o)$ and $(25^o, 10^o)$, respectively. The sensors are arranged in a square array of 5×5 sensors, and are all equal to each other and omnidirectional. Their interdistance is taken to be $\lambda/4$, where λ is the wavelenght of the signals. The number of samples is $N = 100$, and all possible sensor triplets are taken into account, resulting in X, Y and Z matrices of size 16×100. For each algorithm, two signal/noise ratios (SNRs) are considered: 54 dB and 40 dB. For the first case, 100 test runs are performed, while for the 40 dB case 400 runs are done.

In the simulation, method 1 is the 'rotational perturbation' method (section 3.2), method 2 is the additive perturbation method with P_1 and P_2 according to equation (10), while in method 3 we take the perturbation $P_1 = 0$ and P_2 as in equation (11). The results are displayed in figures 1-3, and some statistics are collected in table 1.

While one must be careful with drawing general conclusions from a single example, the following remarks can be made. From figure $1(a)$ the fact that the 'rotational perturbation' method computes the eigenvalues of E_1 and E_2 independently is reflected in the circular shape of the variance clouds. In contrast, the variance clouds in figure 2 and 3 exhibit the line structure of the source configuration, which shows that in the additive perturbation methods the azimuth and elevation directions are not treated independently, and that these methods tend to enhance the source configuration in their estimates.

Finally, when the noise level is increased, method 1 starts to fail (figure $1(b)$), apparently because the assumption that R_2' in equation (8) is 'almost' upper triangular is no longer satisfied. The effect of the increase in noise level on method 2 results in a larger variance in a diagonal direction, while method 3 has a larger variance in vertical directions ('larger' is in comparison with the 1D variance of the eigenvalues of unperturbed matrices, *e.g.,* with respect to the variance of parameters in azimuth direction in figure $3(b)$). This effect becomes more pronounced when the noise level is further increased.

Table 1. Estimate statistics

	SNR = 54 dB (100 runs)				SNR = 40 dB (400 runs)			
	mean(ϕ_i)	mean(θ_i)	std(ϕ_i)	std(θ_i)	mean(ϕ_i)	mean(θ_i)	std(ϕ_i)	std(θ_i)
Method 1	10.00	24.98	0.2	0.2	9.66	24.00	1.3	3.4
	15.01	20.00	0.5	0.6	14.66	19.50	2.2	3.8
	19.98	15.01	0.5	0.5	20.39	15.47	2.1	3.9
	25.02	9.98	0.2	0.2	25.35	11.14	1.4	4.0
Method 2	9.96	24.99	0.7	0.7	9.07	25.94	2.4	2.9
	14.98	19.97	0.7	0.7	15.14	19.93	2.2	2.4
	19.98	14.99	0.7	0.7	19.90	15.24	2.1	2.2
	25.05	9.92	0.8	0.7	26.04	9.10	3.0	2.5
Method 3	10.00	24.98	0.2	0.3	9.66	25.23	1.3	1.9
	14.99	19.98	0.5	0.5	14.66	20.34	2.2	1.8
	19.90	15.06	0.5	0.4	20.39	14.73	2.1	1.8
	25.01	9.94	0.2	0.3	25.35	9.94	1.4	1.9

References

[1] R. Roy, A. Paulraj, and T. Kailath, "ESPRIT—A Subspace Rotation Approach to Estimation of Pa-

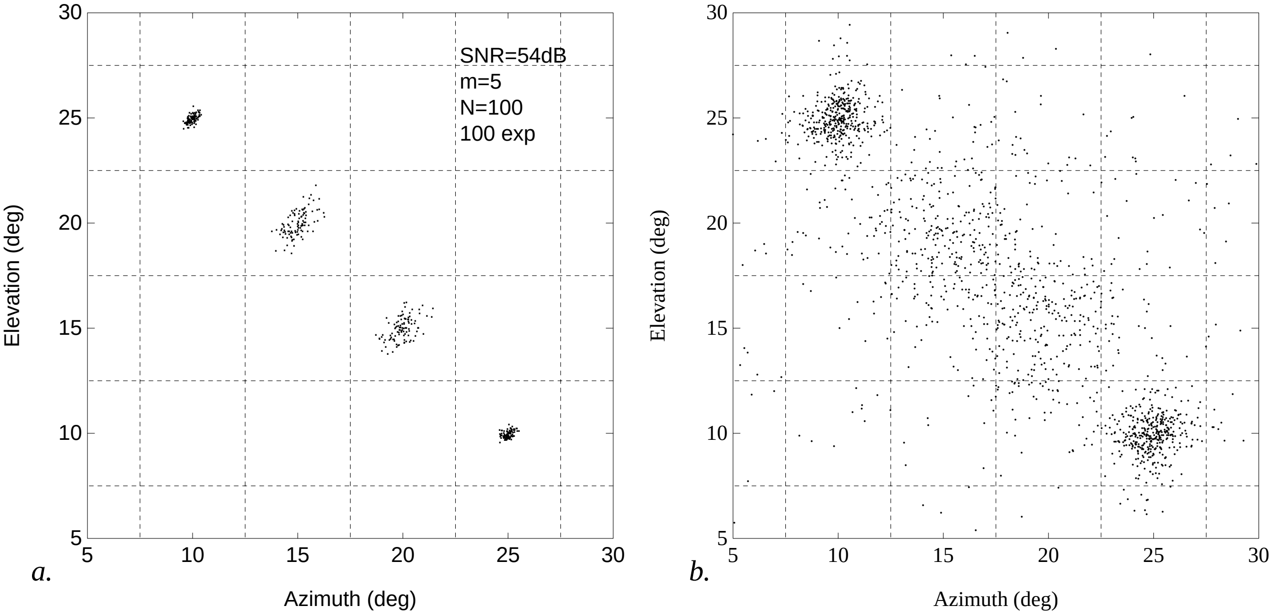

Figure 1. Method 1: (*a*) SNR = 54 dB, (*b*) SNR = 40 dB.

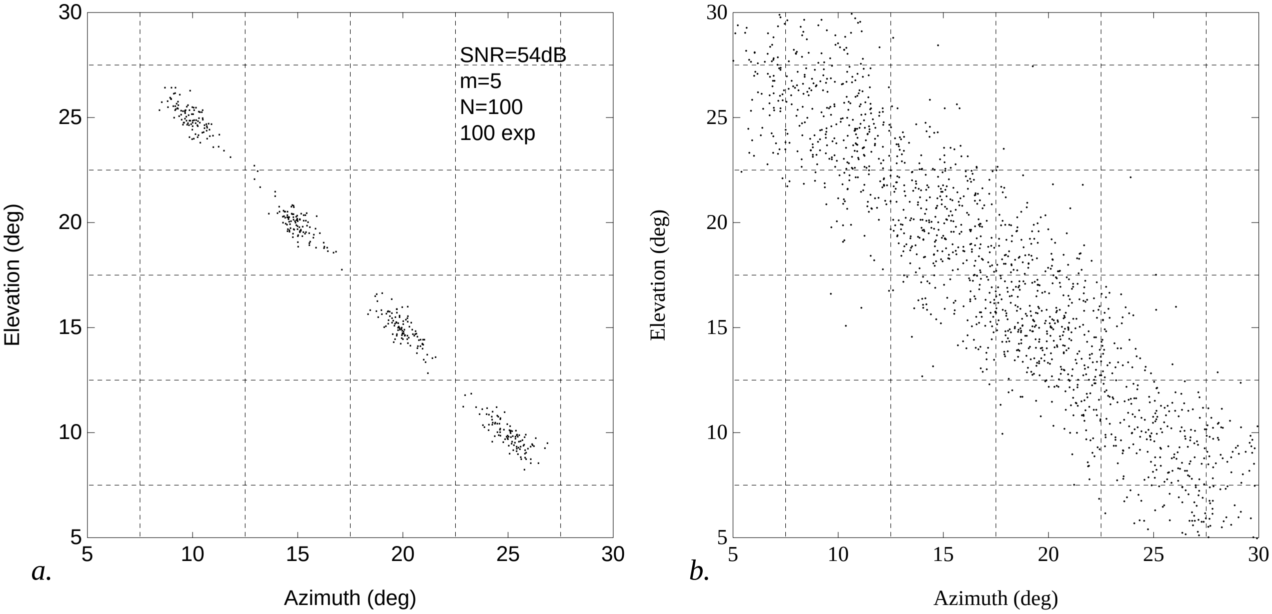

Figure 2. Method 2: (*a*) SNR = 54 dB, (*b*) SNR = 40 dB.

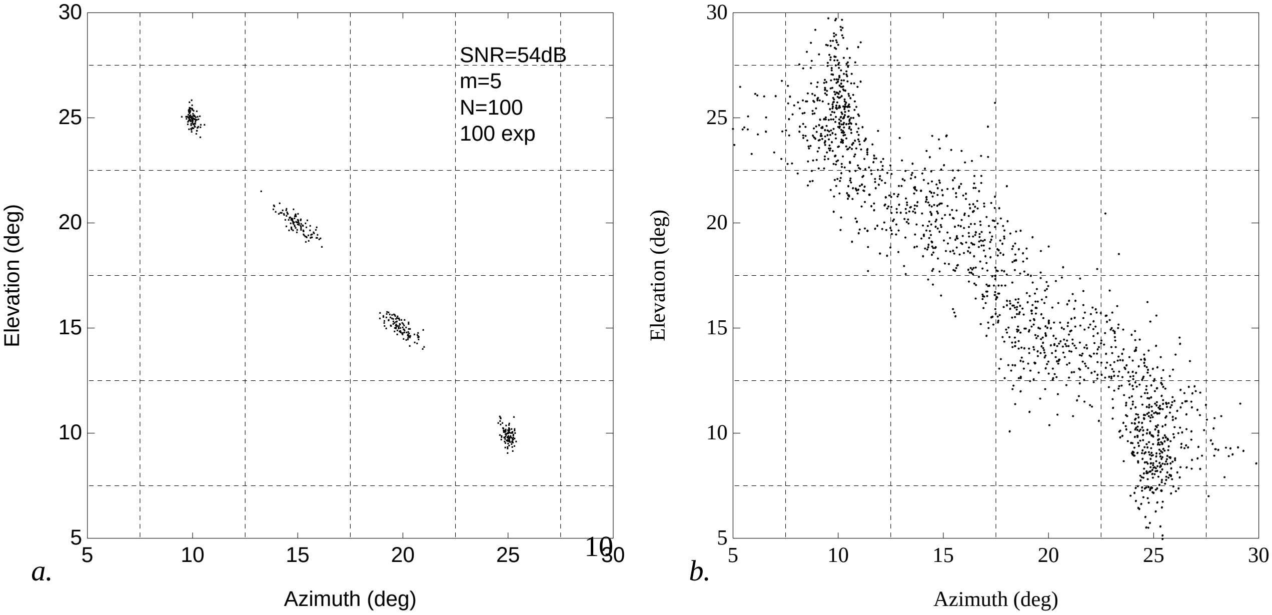

Figure 3. Method 3: (*a*) SNR = 54 dB, (*b*) SNR = 40 dB.

rameters of Cisiods in noise," *IEEE Trans. ASSP*, pp. 1340–1342, 1986.

[2] F. Li and R.J. Vaccaro, "Analytical Performance Prediction of Subspace-Based Algorithms for DOA Estimation," in *Proc. 2nd Int. Workshop on SVD and Signal Processing*, pp. 161–168, Kingston, Rhode Island, June 1990.

[3] A.J. van der Veen and Ed. F. Deprettere, "Parallel VLSI Matrix Pencil Algorithm for High Resolution Direction Finding," *IEEE Trans ASSP*, vol. 39, no. 2, 1991.

[4] A.J. van der Veen, E.F. Deprettere, and A.L. Swindlehurst, "SVD-Based Estimation of Low Rank System Parameters," in *Algorithms and Parallel VLSI Architectures* (E.F. Deprettere and A.J. van der Veen, eds.), Elsevier, 1991.

[5] M.D. Zoltowski and D. Stavrinides, "Sensor Array Signal Processing Via a Procrustus Rotations based Eigenanalysis of the ESPRIT Data Pencil," *IEEE Transactions on ASSP*, vol. 37, no. 6, June 1989.

[6] P.F. Krekel and E.F. Deprettere, "A Two-Dimensional Version of the Matrix Pencil Method to Solve the DOA Problem," *Proc. ECCTD*, pp. 435–439, 1989.

[7] P.B. Ober, E.F. Deprettere, and A.J. van der Veen, "Efficient Methods to Compute Azimuth and Elevation in High-Resolution DOA Estimation," in *Proc. IEEE ICASSP*, (Toronto), 1991.

[8] A.L. Swindlehurst and T. Kailath, "2D Parameter Estimation Using Arrays with Multi-Dimensional Invariance Structure," *Proc. 23-th Asilomar Conf on Signals, Systems and Computers*, pp. 950–954, 1989.

[9] R. Roy, *ESPRIT*. PhD thesis, Stanford Univ., Stanford, CA, 1987.

[10] G. Golub and C.F. Van Loan, *Matrix Computations*. The Johns Hopkins University Press, 1984.

[11] A.L. Swindlehurst and T. Kailath, "Algorithms for Azimuth/Elevation Direction Finding using Regular Array Geometrics," *submitted to IEEE Aerospace and Electronic Systems*, May 1991.