EFFICIENT METHODS TO COMPUTE AZIMUTH AND ELEVATION IN HIGH RESOLUTION DOA ESTIMATION †

P. Bas Ober, Ed F. Deprettere and Alle-Jan van der Veen
Department of Electrical Engineering
Delft University of Technology
2628 CD Delft, The Netherlands

In this paper, a number of high-resolution methods are discussed to determine the two-dimensional (α_i, β_i) directions of arrival (DOA) of a number of plane waves, impinging on a sensor array. The array possesses in general an arbitrary geometry, but consists of triplets or quadruplets of sensors that are identical. The data from these sensors is first processed for each independent dimension using an ESPRIT-like subspace based method. The correct (α_i, β_i) pairs are then determined by considering subspace properties that the processed data would possess in the noise free case. A new algorithm is devised that potentially improves the parameter estimates and yields the correct parameter pairs while avoiding an extensive search over the two separate one-dimensional parameter sets. In addition, resolution is improved by exploiting the spatial dependency of the measurements when sensor quadruplets are used.

1. INTRODUCTION

Subspace based high resolution DOA algorithms considered in this paper [1, 2, 3] 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 1D problems provides us with two decoupled parameter sets, that have to be combined again to give the correct parameter pairs. In one approach of the 1D DOA problem [1], a matrix pencil is formed, the general eigenvalues of which (implicitly) contain the directions of arrival of impinging signals. In the 1D based 2D case, two pencils are constructed on three data matrices, and, after solving these pencils independently, a problem of eigenvalue matching occurs. An algorithm must be found to combine azimuths and elevations correctly. Different approaches have been alluded to in the literature:

- Correlation technique: Described in Zoltowski et al. [4]. After calculating the correlation matrix between the outputs of all sensors, compute its eigenvectors that correspond to the noise. The correct pairing is labeled as that for which the array manifold vector projection on the noise-subspace is 'closest to zero', followed by a MUSIC search in parameter space if desired. This method is computationally not very attractive.
- Overdetermined direction data: 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 [4].

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 [5] for the 1D case.

In this paper, a new and different approach is considered, which is to approximate the data matrices in such a way that their eigenvectors become equal, as would be the case when no noise is present. In that case, both matrices can be diagonalized by the same similarity transform then, and the two pencils that are formed can be solved simultaneously. The underlying property was already mentioned in [6] and exploited there in some sense, combined with spacial overdetermining to improve the resolution and correct pairing results. The correct pairings are found directly. The approximation of the data matrices can also be done by computing the

eigenvectors of both data matrices, searching for 'near' eigenvectors, and projecting these vector pairs onto a single best-fitting third. Again, a search has to be performed. Yet another approach is based on the assumption that all noise is purely additive. The purpose is cancel this noise, at least partially, by adding small perturbation matrices to the obtained matrices, such that their eigenvectors will be equal afterwards, or, equivalently, that the resulting matrices will commute. This last algorithm is considered to be the main contribution of this paper.

In section 2, a short review of the ESPRIT data model is presented. In section 3, the corresponding matrix pencil approach is reviewed. Section 4 deals with the properties of the data in the noise free case. In section 5 a number of approaches to enforce these properties on noisy data are discussed, in context with a review of some other algorithms. This gives rise to a new algorithm, whose promising numerical properties yet have to be investigated.

2. 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 $\vec{d}_{xy}$ and $\vec{d}_{xz}$ between its components are required to be the same (and, for convenience, are assumed to be orthogonal). This way, assigning one sensor of each triplet to one of the sensor-arrays X, Y and Z , three identical although displaced arrays are obtained. 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 :

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

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

$$x_i(t) = \sum_{k=1}^d a_{ik} s_k(t) + n_{xi}(t)$$

$$y_i(t) = \sum_{k=1}^d a_{ik} \phi_k s_k(t) + n_{yi}(t)$$

$$z_i(t) = \sum_{k=1}^d a_{ik} \theta_k s_k(t) + n_{zi}(t)$$

where a_{ik} is the gain of sensor i for signal $s_k(t)$, n_{xi} , n_{yi} , n_{zi} represent noise, 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 $\vec{u}_k$ is the direction of propagation of $s_k(t)$, they can be written as

$$\phi_k = e^{-j \frac{\omega_0}{c} (\vec{d}_{xy}^T \vec{u}_k)} \quad (2.1a)$$

$$\theta_k = e^{-j \frac{\omega_0}{c} (\vec{d}_{xz}^T \vec{u}_k)} \quad (2.1b)$$

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

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

$$\begin{aligned} X &= AS + N_x \\ Y &= A\Phi S + N_y \\ Z &= A\Theta S + N_z \end{aligned} \quad (2.2)$$

where $X(i,j) = x_i(t_j)$, $Y(i,j) = y_i(t_j)$, $Z(i,j) = z_i(t_j)$, $A(i,k) = a_{ik}$, $S(k,j) = s_k(t_j)$ and $N_x(i,j) = n_{xi}(t_j)$, $N_y(i,j) = n_{yi}(t_j)$, $N_z(i,j) = n_{zi}(t_j)$. 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 ($d \times N$). A and S are unknown. The matrices Φ and Θ are diagonal and contain the phase shifts (2.1) for each signal:

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

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

3. THE MATRIX PENCIL APPROACH

Matrix polynomials of the form $E - \alpha F$, $\alpha \in C$, 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 $\mu = \mu_j$, $i, j = 1, 2, \dots, d$, 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 [7] of the data matrices 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 following closely follows the approach outlined in [3].

Compute the SVD's of two matrices, constructed from X, Y and Z :

$$\begin{bmatrix} X & Y & Z \end{bmatrix} = U_1 \Sigma_1 V_1^H = U_1 \Sigma_1 \cdot \begin{bmatrix} V_{11}^H & V_{21}^H & V_{31}^H \end{bmatrix} \quad (3.1a)$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = U_2 \Sigma_2 V_2^H \quad (3.1b)$$

where U_1 and V_1 are unitary matrices, and Σ_1 is a diagonal matrix containing the singular values. Then 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

$$\begin{bmatrix} X & Y & Z \end{bmatrix} = \hat{U}_1 \hat{\Sigma}_1 \hat{V}_1 \quad (3.2a)$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \hat{U}_2 \hat{\Sigma}_2 \hat{V}_2 \quad (3.2b)$$

in which $\hat{U}_1, \hat{V}_1$ contain the columns of the U_1, V_1 that correspond to the nonzero singular values in $\hat{\Sigma}_1$. The TLS approximations of the data matrices share the same column space $\text{range}(\hat{U}_1)$ and row space $\text{range}(\hat{V}_2)$, as explicitly determined in (3.2), and are obtained by projecting X, Y and Z onto these subspaces:

$$\begin{aligned} \hat{X} &= (\hat{U}_1 \hat{U}_1^H) X (\hat{V}_2 \hat{V}_2^H) = \hat{U}_1 \hat{E}_x \hat{V}_2^H \\ \hat{Y} &= (\hat{U}_1 \hat{U}_1^H) Y (\hat{V}_2 \hat{V}_2^H) = \hat{U}_1 \hat{E}_y \hat{V}_2^H \\ \hat{Z} &= (\hat{U}_1 \hat{U}_1^H) Z (\hat{V}_2 \hat{V}_2^H) = \hat{U}_1 \hat{E}_z \hat{V}_2^H \end{aligned}$$

in which $\hat{E}_x, \hat{E}_y$ and $\hat{E}_z$ are square $d \times d$ matrices, and with

$$\begin{aligned} \hat{E}_x &= \hat{U}_1^H \hat{X} \hat{V}_2 = \hat{\Sigma}_1 \hat{V}_{11}^H \hat{V}_2 \\ \hat{E}_y &= \hat{U}_1^H \hat{Y} \hat{V}_2 = \hat{\Sigma}_1 \hat{V}_{21}^H \hat{V}_2 \\ \hat{E}_z &= \hat{U}_1^H \hat{Z} \hat{V}_2 = \hat{\Sigma}_1 \hat{V}_{31}^H \hat{V}_2 \end{aligned} \quad (3.3)$$

and

$$\hat{V}_1^H = \begin{bmatrix} \hat{V}_{11} & \hat{V}_{21} & \hat{V}_{31} \end{bmatrix}$$

The generalized eigenvalues $\text{GE}(E_y, E_x)$, or the eigenvalues of $E_x^{-1} E_y$, are equal to the rank reducing numbers of the matrix pencil $X - \lambda Y$. The same holds for the generalized eigenvalues of (E_z, E_x) and the pencil $X - \lambda Z$. To determine these eigenvalues from the E_i , multiplication by $\hat{\Sigma}_1$ 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 [5].

Substituting (2.2) in (3.3) and defining $A_e = \hat{U}_1^H A$ and $S_e = \hat{S} \hat{V}_2$, we arrive at

$$E_x = A_e S_e + N_{x,e} \quad (3.4a)$$

$$E_y = A_e \Phi S_e + N_{y,e} \quad (3.4b)$$

$$E_z = A_e \Theta S_e + N_{z,e} \quad (3.4c)$$

where $N_{x,e} = \hat{U}_1^H N_x \hat{V}_2$ etc. In order to improve resolution properties yet another subarray, say W , can be used, having a displacement vector $\vec{d}_{xw} = \vec{d}_{xy} + \vec{d}_{xz}$ (instead of triplets, groups of four matched sensors are needed, which will require only one more sensor if rectangular, equally spaced overlapping arrays are used). The decomposition structure of the fourth data matrix is

$$W = A\Phi\Theta S + N_w$$

Replacing the matrices in (3.2) by $\{X \ Y \ Z \ W\}$ and $\{X^H \ Y^H \ Z^H \ W^H\}$ and doing the same type of calculations, we arrive at

$$E_w = A_e \Phi \Theta S_e + N_{w,e} \quad (3.4d)$$

4. PROPERTIES OF E_x, E_y AND E_z

From (3.4) can directly be derived that, without noise, the matrix E_a , defined by

$$E_a = \begin{bmatrix} E_x & E_y \\ E_z & E_w \end{bmatrix} = \begin{bmatrix} A_e \\ A_e \Theta \end{bmatrix} \begin{bmatrix} S_e & \Phi S_e \end{bmatrix} \quad (4.1)$$

obviously has rank d .

When we define

$$E_1 = E_x^{-1} E_y = S_e^{-1} \Phi S_e \quad (4.2a)$$

$$E_2 = E_x^{-1} E_z = S_e^{-1} \Theta S_e \quad (4.2b)$$

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 : if

$$Q_1^H E_1 Q_1 = Q_1^H S_e^{-1} \Phi S_e Q_1 = R_1 \quad (4.3a)$$

$$Q_2^H E_2 Q_2 = Q_2^H S_e^{-1} \Theta S_e Q_2 = R_2 \quad (4.3b)$$

then $Q_1 = Q_2 = Q$, and the upper triangular matrices R_1 and R_2 have diagonals equal to $\text{diag}(\Phi)$ and $\text{diag}(\Theta)$, 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 'permutation-less rotation' approach in section 5.2.

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

5. A BRIEF TOUR OF PAIR MATCHING ALGORITHMS

5.1 The Searching Algorithm[4]

In this algorithm, reported by Zoltowski et al. and which we will name 'Searching algorithm', three instead of two pencils are formed:

$$\begin{aligned} X - \lambda Y &= A(I - \lambda \Phi)S + \text{noise} \\ X - \mu Z &= A(I - \mu \Theta)S + \text{noise} \\ Y - \nu Z &= A(\Phi - \nu \Theta)S + \text{noise} \end{aligned}$$

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'

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

where $\text{diag}(R_1)$ contains the estimated ϕ_i , $\text{diag}(R_2)$ contains the estimated θ_i and $\text{diag}(R_3)$ the quotients ϕ_i/θ_i , like in (4.3), using an extra $E_3 = E_1^T E_2$.

5.2 Matching using permutation-less rotation matrices[6]

Experience gained with the Jacobi-iteration method for eigenvalue calculations [5] led to the following pairing algorithm, which can be integrated with the eigenvalue computations needed anyway.

Triangularize E_1 as in (4.3) (in practise, do this with Generalized Eigenvalues): determine Q_1 to triangularize E_1 , and simultaneously apply the same similarity transform to E_2 :

$$\begin{aligned} Q_1^H E_1 Q_1 &= R_1 \\ Q_1^H E_2 Q_1 &= R_2' \end{aligned}$$

in which R_2' shall 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' completely upper triangular; i.e. Q_2' such that

$$Q_2'^H R_2' Q_2' = R_2$$

is a unitary matrix which is almost equal to the identity matrix. This means that Q_2' does not permute the rough eigenvalue estimates in R_2' while computing the correct eigenvalues. 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 same as obtained in e.g. Zoltowski's method, but this time, we get the correct pairings 'for free'.

5.3 Forcing E_a To Have Rank d

We now briefly consider how a fourth data matrix W , or E_w , can be exploited in the previous two algorithms in order to improve resolution and accuracy. As stated in (4.1), the matrix E_a defined by

$$E_a = \begin{bmatrix} E_x & E_y \\ E_z & E_w \end{bmatrix}$$

would be of rank d in the noise free situation. This property is lost due to noise, but can be restored by computing the SVD of E_a and setting the d smallest singular values to zero. A rank d approximation $\hat{E}_a$ will result, with

$$\hat{E}_a = \begin{bmatrix} \hat{E}_x & \hat{E}_y \\ \hat{E}_z & \hat{E}_w \end{bmatrix}$$

The submatrices $\hat{E}_x, \hat{E}_y, \hat{E}_z$ can be used instead of E_x, E_y and E_z in the previous two algorithms, and correspondingly, the eigenvalues of $E_1 = \hat{E}_x$ and $E_2 = \hat{E}_z$ must be determined.

5.4 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 :

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

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 (2.1). It is not possible to find the correct tuples $\{\phi_k, \theta_k\}$ 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

$$\begin{aligned} (E_1 + P_1)(E_2 + P_2) &= (E_2 + P_2)(E_1 + P_1) \\ \Leftrightarrow (E_1 E_2 - E_2 E_1) + (P_1 P_2 - P_2 P_1) &= (E_2 P_1 - P_1 E_2) - (E_1 P_2 - P_2 E_1) \\ \Leftrightarrow (E_1 E_2 - E_2 E_1) + (P_1 P_2 - P_2 P_1) &= \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \begin{bmatrix} E_2 \\ -E_1 \end{bmatrix} \end{aligned} \quad (5.1a)$$

and such that P_1, P_2 obey some minimum norm constraint, e.g.

$$\text{Minimize } \|P_1\|_F^2 + \|P_2\|_F^2 \quad (5.1b)$$

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. An easier to compute approximate solution for P_1 and P_2 can however be obtained by neglecting the term $P_1 P_2 - P_2 P_1$, 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, e.g., by iteration, eigenvalue computations or Kronecker products. Evaluating this last method, we arrive at the following equations:

$$\begin{bmatrix} -E_1^T \oplus E_2 & E_1^T \oplus -E_1 \end{bmatrix} \cdot \text{vec} \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} = \text{vec}(E_1 E_2 - E_2 E_1)$$

from which we can find, using the Moore-Penrose pseudo-inverse denoted by $^{+}$, the solution with minimum perturbation norm,

$$\text{vec} \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} = \begin{bmatrix} -E_1^T \oplus E_2 & E_1^T \oplus -E_1 \end{bmatrix}^{+} \cdot \text{vec}(E_1 E_2 - E_2 E_1)$$

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, 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. 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. The linearity of the constraint ensures that there is, indeed, a solution of minimum norm.

6. CONCLUDING REMARKS

While a thorough simulation study still needs to be done, preliminary simulations show that overdetermination by a fourth data matrix W seems to be a useful tool for improving resolution. It can be used in combination with all methods. Zoltowski's searching method and the 'permutation-less rotation' method arrive at the same parameter values. The preliminary simulations show that the pairing also is the same, but this must still be confirmed for low SNR's. Our new perturbation method seems to be as accurate as the above two methods, and performs even slightly better if we look at the Mean Square Errors in the computations of the angles. This is true even though only an approximate solution of problem (5.1) is used. The assumption that the term $P_1 P_2 - P_2 P_1$ can be neglected therefore doesn't seem to be an unreasonable one. It remains yet to be verified that this potential improvement is significant in view of the extra computations needed. Finally, while the new algorithm calls for the solution of only one eigenvalue problem, a more efficient algorithm to arrive at the correct perturbation matrices, avoiding inversion of unwieldy Kronecker matrices, definitely should and most probably can be devised.

References

1. R. Roy, A. Paulraj, and T. Kailath, "ESPRIT-A Subspace Rotation Approach to Estimation of Parameters of Cissoids in noise", *IEEE Trans. ASSP* 34(1986).
2. F. Li and R. Vaccaro, "Analytic Performance Prediction of Subspace-Based Algorithms for DOA Estimation", *Proc. Int. Workshop SVD and Signal Processing, Rhode Island*, pp. 161-168 (June 1990).
3. A.J. van der Veen, E.F. Deprettere, and A.L. Swindlehurst, "SVD-Based Approximation of Low-Rank System Parameters", in *Algorithms and Parallel VLSI Architectures*, ed. E.F. Deprettere & A.J. van der Veen, (1991).
4. M.D. Zoltowski and D. Stavrinides, "Sensor Array Signal Processing Via a Procrustes Rotations based Eigenanalysis of the ESPRIT Data Pencil", *IEEE Transactions on ASSP* 37(6)(June 1989).
5. A.J. van der Veen and E.F. Deprettere, "Parallel VLSI Matrix Pencil Algorithm for High Resolution Direction Finding", *IEEE Trans. ASSP* 39(2)(Feb. 1991).
6. P.F.C. Krekel and E.F. Deprettere, "A Two-Dimensional Version of the Matrix Pencil Method to Solve the DOA Problem", *Proc. ICASSP*, pp. 435-439 (Sept. 1988).
7. G.H. Golub and C.F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press (1983).