References


**Abstract.** Centralized methods for source location using sensor arrays have computational and communication burdens that increase significantly with the number of sensors in the array. Therefore, these methods may not be usable in the applications involving very large arrays. In such applications, the data processing may need to be decentralized. This paper introduces two methods for decentralized array processing, based on the recently proposed MODE algorithm. For prescribed nonoverlapping subarrays, both methods are shown to be statistically optimal in the sense that asymptotically they provide the most accurate decentralized estimates of source location parameters. The problem of subarray selection to further optimize the estimation accuracy is only briefly addressed. The two methods are intended for different types of applications: the first should be preferred when there exist significant possibilities for local processing or for parallel computation in the central processor; otherwise the second method should be preferred. The accuracy of the two decentralized methods is compared to the centralized Cramér–Rao bound, both analytically and numerically, in order to provide indications about the loss of accuracy associated with decentralized processing.

**1. Introduction**

Source location using arrays of sensors is a topic of considerable importance for a number of applications such as sonar, radar, microwave communications,

* Received March 15, 1992; accepted September 23, 1992. The work of P. Stoica was supported by a grant from the Royal Swedish Academy of Engineering Sciences and by the Swedish Research Council for Engineering Sciences under contract 91-676. The work of A. Nehorai was supported by the Air Force Office of Scientific Research under grant no. AFOSR-90-0164, and by the Office of Naval Research under grant no. N00014-91-J-1298.

1 Automatic Control Department, Bucharest Polytechnic Institute, Splaiul Independenței 313, R-77206 Bucharest, Romania, on leave at the Department of Applied Electronics, Chalmers University of Technology, 2-41296 Göteborg, Sweden.

2 Department of Electrical Engineering, Yale University, New Haven, CT 06520.

3 Automatic Control and Systems Analysis Group, Department of Technology, Uppsala University, P.O. Box 27, S-751 03 Uppsala, Sweden.
statistics optimal in the class of decentralized estimates, in the sense that they provide minimum-variance DOA estimates in this class. The two DEMODE's aim at different types of applications: DEMODE-1 should be preferred when there exist significant possibilities for local processing or for parallel computation in the central processor; otherwise DEMODE-2 should be preferred.

As implied by the previous discussion, the covariance matrix of the DOA estimation errors associated with DEMODE-1 or DEMODE-2 is a lower bound on the covariance matrix corresponding to any other decentralized DOA method in a fairly general class. In this paper, we also compare this lower bound with the Cramér–Rao bound (CRB) for centralized processing, in order to study the loss of accuracy that is inherently associated with decentralized processing. The influence of the subarray selection on this loss of performance associated with decentralized processing is also briefly studied. Another tradeoff associated with decentralized processing is its lower resolution. More specifically, the centralized processor may resolve all the sources that are present in the array field, whereas the decentralized processor may fail to do so. As already stated, we assume that the number of sources is known to the processor. Therefore we do not address the "resolution tradeoff" in this paper.

2. Array model

Consider a situation in which an array of $m$ sensors is situated in the wavefield generated by $n$ narrowband far-field sources. Assume for simplicity that the array and the sources belong to the same plane. Let $a(\theta)$ denote the response of the array to a unit–amplitude waveform impinging from the direction $\theta$. If the locations and the attenuation characteristics of the sensors composing the array are assumed to be known, then $a(\cdot)$ is a function of the direction-of-arrival (DOA) $\theta$ only. Under the above assumptions, it is well known ([1], [4], [10], [13]–[15]) that the array output can be modelled by the following equation:

$$y(t) = Ax(t) + e(t)$$ (2.1)

where $y(t)$ denotes the vector of sensor outputs, $x(t)$ is the vector of source signals, $e(t)$ is an additive noise, and

$$A = [a(\theta_1) \ldots a(\theta_n)]$$

The following standard assumptions on the model (2.1) are considered to hold throughout the paper.

A1. $m > n$, and the vectors $\{a(\theta_i)\}$ corresponding to any set of $(n + 1)$ distinct DOA's are linearly independent.

1 In fact, it is assumed that $m \gg n$ to motivate the need for decentralized processing.
A2. The source signals are stationary ergodic random variables. Moreover, they have a nonsingular covariance matrix
\[ P = E x(t)x^*(t) \quad \det P \neq 0 \] (2.2)
(where “*” denotes the conjugate transpose and “det” is a short notation for determinant). Note that the signals \( x(t) \) are not required to be temporally white and Gaussian distributed, as is usually assumed in the array processing literature.

A3. The noise \( e(t) \) is Gaussian distributed with zero mean and the following covariances:
\[ E e(t)e^*(s) = \sigma I \delta_{t,s} \] (2.3)
\[ E e(t)e^T(s) = 0 \quad \text{for all } t \text{ and } s. \] (2.4)

(Here “T” denotes the transpose, and \( \delta_{t,s} \) is the Kronecker delta.) Thus, \( e(t) \) is assumed to be uncorrelated both temporally and spatially. It is also assumed that \( e(t) \) is independent of \( x(s) \), for all \( t \) and \( s \).

The problem is to estimate the DOA vector:
\[ \omega = [ \theta_1 \ldots \theta_n ]^T \] (2.5)
from \( N \) samples of the sensor output vector \( (y(1), \ldots, y(N)) \). (Note that we chose to denote the DOA vector by \( \omega \); the symbol \( \theta \) will be used to denote a general element of \( \omega \).) The solutions (both centralized and decentralized) to the above problem, which will be presented in the following sections, make use of a number of basic facts on the eigenstructure of the covariance matrix of the output vector. These facts are collected, for easy reference, at the end of this section.

Let \( R = E y(t)y^*(t) \).

It follows easily from (2.1) and assumptions A2 and A3 that
\[ R = APA^* + \sigma I. \] (2.7)

Let \( \{ \lambda_k \}_{k=1}^n \) denote the eigenvalues of \( R \), arranged in decreasing order \( \lambda_1 \geq \ldots \geq \lambda_n \), and let \( [s_k]_{k=1}^n \) denote the corresponding orthonormal eigenvectors. Since \( \text{rank}(APA^*) = n \) by assumptions A1 and A2, it is easy to see from (2.7) that
\[ \lambda_k > \sigma \quad (k = 1, \ldots, n) \] (2.8)
and
\[ \lambda_k = \sigma \quad (k = n+1, \ldots, m). \] (2.9)

Define
\[ \Lambda = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix} \] (2.10)

Using this notation one can write
\[ R = SAS^* + \sigma GG^* = S \hat{\Lambda} S^* + \sigma I. \] (2.14)

From the two equations above for \( R \), (2.7) and (2.14), it follows that
\[ \text{range}(A) = \text{range}(S) \] (2.15)
\[ A^*G = 0 \] (2.16)
\[ A(A^*A)^{-1}A^* = SS^* = I - GG^*. \] (2.17)

The sample counterparts of the previously introduced quantities are denoted by a "caret." Thus, the sample covariance matrix is
\[ \hat{R} = \frac{1}{N} \sum_{i=1}^N y(t)y^*(t) \] (2.18)

and \( \hat{\Lambda}, \hat{\Lambda}, \hat{S}, \text{ and } \hat{G} \) denote the matrices \( \Lambda, \hat{\Lambda}, S \) and \( G \) made of the eigenvectors of \( \hat{R} \).

3. Centralized MODE

The centralized MODE estimate of the DOA vector \( \omega \) is given by, see [10],
\[ \hat{\omega} = \arg \min \omega \ h(\omega) \] (3.1)
with
\[ h(\omega) = \text{tr} \left[ A^*(\omega) \hat{G} \hat{G}^* A(\omega) \hat{W} \right] \] (3.2)
where \( \text{tr} \) denotes the trace operator, and \( \hat{W} \) is a consistent estimate of the following matrix,
\[ W = \left[ A^*S \Lambda^{-2} \Lambda S^*A \right]^{-1} = \left[ P^{-1} + \sigma P^{-1}(A^*A)^{-1}P^{-1} \right]^{-1}. \] (3.3)

The equivalence of the above two expressions for \( W \) is established in [8].

The main results concerning the statistical properties of \( \hat{\omega} \) are presented in the following theorem.

Theorem 3.1 The estimate \( \hat{\omega} \) given by (3.1) is Gaussian distributed in large samples \( (N \gg 1) \) with mean \( \omega \) and covariance matrix
\[ C_{CR} = \frac{\sigma}{2N} \left[ \text{Re}[H \odot W^T] \right]^{-1} \] (3.4)
where $\text{Re}\{\cdot\}$ denotes the real part of the bracketed quantity, $\odot$ stands for the Hadamard matrix product (i.e., elementwise multiplication), and

$$H = D^*GG^*D. \quad (3.5)$$

In (3.5), $D$ is the following matrix

$$D = [d_1 \ldots d_n] \quad (3.6)$$

where

$$d_k = \frac{\partial a(\theta)}{\partial \theta} \bigg|_{\theta = \theta_k}. \quad (3.7)$$

Furthermore, if $x(t)$ is Gaussian distributed and temporally white, then $C_{CR}$ is the Cramér–Rao lower bound on the covariance matrix of any consistent estimate of the DOA vector $\omega$.

**Proof.** The stated distributional properties of $\hat{\omega}$ have been established in [10]. The fact that the asymptotic covariance matrix $C_{CR}$ of $\hat{\omega}$ coincides with the CRB has been proved in [7] (see also [13]). In Appendix A, we present a new proof of expression (3.4) for the covariance matrix of $\hat{\omega}$. This proof is more general than the previously available proof (it does not make use of a number of assumptions on $x(t)$, as pointed out in Section 2), and it is simpler, too. However, the main virtue of this new proof is that, unlike the original proof, it extends readily to the case of decentralized processing (a fact exploited in the following sections).

It follows from Theorem 3.1, that the centralized MODE estimate is asymptotically statistically efficient in the sense that (under the Gaussian hypothesis) no other method can provide more accurate DOA estimates, in large samples. The implementation of this estimate can be done in the following steps.

**Centralized MODE algorithm**

**Step 1.** Compute the sample covariance matrix $\hat{R}$ and its eigendecomposition.

**Step 2.** Use the eigenvectors of $\hat{R}$ to determine the number of sources $n$ by one of the techniques described in [14], [15].

**Step 3.** Determine an initial estimate $\hat{\omega}^0$ of $\omega$ by minimizing the loss function (3.2) with $\hat{W} = I$. Since

$$h(\omega)_{\hat{W} = I} = \text{tr}[A^* \hat{G}\hat{G}^*A] = \sum_{k=1}^{n} a^*(\theta_k)\hat{G}\hat{G}^*a(\theta_k) \quad (3.8)$$

it readily follows that the elements of $\hat{\omega}^0$ are given by the locations of the global minimum points of the following function,

$$g(\theta) = a^*(\theta)\hat{G}\hat{G}^*a(\theta) = a^*(\theta)a(\theta) - a^*(\theta)\hat{S}\hat{S}^*a(\theta). \quad (3.9)$$

Thus, $\hat{\omega}^0$ is nothing but the MUSIC estimate of $\omega$ [4],[8],[14]!

The minimization of $g(\theta)$ can be done by evaluating this function at the points of a fine grid over the domain of interest. Note that the second expression of $g(\theta)$ in (3.9) is computationally more convenient than the first when $m > 2n$. Note also that some care in defining $\hat{\omega}^0$ should be exercised in the cases where $g(\theta)$ has less than $n$ dominant minima. (In such cases the location of one minimum point may be used as an initial estimate for two or more elements of $\omega$.)

**Step 4.** Estimate $W$ as

$$\hat{W} = \left[ [A^*\hat{S}\hat{S}^*A]^{-1} \right]_{\omega = \hat{\omega}^0}. \quad (3.10)$$

**Step 5.** Determine the MODE estimate of $\omega$ by minimizing the function $h(\omega)$ defined by (3.2) where $\hat{W}$ is as computed in Step 4. The minimization of this function can be conveniently done by using the following Gauss–Newton algorithm:

$$\hat{\omega}^{k+1} = \hat{\omega}^k - \left[ \frac{\partial h(\omega)}{\partial \hat{W}} \right]_{\omega = \hat{\omega}^k}^{-1} \left[ \frac{\partial h(\omega)}{\partial \mu} \right]_{\omega = \hat{\omega}^k} \quad (k = 0, 1, 2, \ldots) \quad (3.11)$$

where the superscript $k$ denotes the iteration number, $\hat{\omega}^0$ is the initial estimate computed in Step 3,

$$\hat{H} = D^*\hat{G}\hat{G}^*D = D^*D - D^*\hat{S}\hat{S}^*D \quad (3.12)$$

and the $k$ element of the vector $\mu$ is given by

$$\mu_k = \left[ D^*\hat{G}\hat{G}^*A\hat{W} \right]_{kk} = \left[ D^*(I - \hat{S}\hat{S}^*)A\hat{W} \right]_{kk} \quad (3.13)$$

In order to motivate (3.11), note that $2\text{Re}[\mu]$ is the gradient vector of $h(\omega)$ (see equation (A.4) in Appendix A), and $2\text{Re}[\hat{H} \odot \hat{W}^T]_{\omega = \hat{\omega}^0}$ is a Gauss-like approximation of the Hessian matrix of $h(\omega)$ (see equation (A.6) and the subsequent discussion in Appendix A). Note that the matrix $\hat{W}$ can be frozen at the value computed in Step 4, or it can be re-evaluated at each iteration using the latest available estimate of $\omega$ (however, re-evaluation of $\hat{W}$ will increase the computational burden without necessarily improving the estimates). It can be shown that, for sufficiently large $N$, the Gauss–Newton algorithm (3.11) would converge in one iteration. In practical applications, one iteration may be insufficient to achieve the convergence of the algorithm, but $2 \div 3$ iterations may suffice.

The number of arithmetic operations required to implement Steps 2 through 5 is proportional to $m$; more exactly this number is $O(n^2m)$ for Steps 2, 4, and 5, and

2 If reduction of the computational burden is essential, $\hat{H}$ could also be frozen at the initial value $[H]_{\omega = \hat{\omega}^0}$. The algorithm (3.11) with $\text{Re}[\hat{H} \odot \hat{W}^T]_{\omega = \hat{\omega}^0}$ replaced by $\text{Re}[\hat{H} \odot \hat{W}^T]_{\omega = \hat{\omega}^0}$ can still be shown to converge to a minimum of the function $h(\omega)$ (see, for example, [5], Chapter 7).
0(pnm) for Step 3, where p is the number of points of the grid used to minimize g(θ). For Step 1, however, the number of corresponding arithmetic operations increases much more rapidly with m: computation of ̂R requires O(Nm^2) operations, and the eigendecomposition O(m^3). For very large arrays containing hundreds of sensors, the computational burden induced by Step 1 may be prohibitively large. In addition, the capacity of the communication channels may be insufficient for transmitting the large number of sensor output data. In such cases, the use of the centralized MODE may not be possible and one has to resort to decentralized processing. The decentralization of the MODE algorithm described above is the subject of the following section.

4. Decentralized MODE with hard local processing requirements

Assume that the array is decomposed into L nonoverlapping subarrays, each of which has a local processor. Assume also that each subarray satisfies assumption A1 introduced in Section 2. Let A_s denote the A-matrix corresponding to subarray s (s = 1, . . . , L). Then,

\[ A = \begin{pmatrix} A_1 \\ \vdots \\ A_L \end{pmatrix}. \]  

(4.1)

The data measured in each subarray can be processed by the MODE algorithm of Section 3 to obtain an estimate ̂ω_s of the DOA vector ω. These estimates \{ ̂ω_s \}_{s=1}^{L} will then be communicated to the central processor (also called the "fusion center") which will combine them in some way. The covariances of the estimates ̂ω_s must play an important role in deriving a statistically motivated combination procedure. These covariances are established in the following.

Theorem 4.1. Let \{ ̂ω_s \} denote the estimate of ω, obtained from subarray s (s = 1, . . . , L) by using the MODE algorithm. Then, for sufficiently large N,

\[ E( ̂ω_μ - ω)( ̂ω_ν - ω)^T = C_μδ_μ,ν \]  

(4.2)

where \( C_μ \) is the covariance matrix (3.4) corresponding to the \( μ \)th subarray.

Proof. See Appendix B.

Remark. Note that the estimates ̂ω_μ and ̂ω_ν are uncorrelated for \( μ ≠ ν \), even though the subvectors \( y_μ(t) \) and \( y_ν(t) \) of \( y(t) \), from which they are computed, are correlated: \( E y_μ(t)y_ν^T(t) = A_μ P A_ν^T \).

It is now possible to derive the following statistically optimal combiner.

Array Processing Using MODE Algorithm

Theorem 4.2. Let ̂C_s be a consistent estimate of the covariance matrix \( C_s \), introduced in Theorem 4.1 (s = 1, . . . , L), and let

\[ ̂ω = \left( \sum_{s=1}^{L} ̂C_s^{-1} \right)^{-1} \left( \sum_{s=1}^{L} ̂C_s^{-1} ̂ω_s \right). \]  

(4.3)

The estimate ̂ω, (4.3), is Gaussian distributed in large samples, with mean ω and the following covariance matrix,

\[ C_{LB} = \left( \sum_{s=1}^{L} C_s^{-1} \right)^{-1}. \]  

(4.4)

Furthermore, \( C_{LB} \) is a lower bound on the asymptotic covariance matrix of any estimate of ω obtained from consistent, uncorrelated, and Gaussian distributed subarray DOA estimates.

Proof. The theorem essentially follows from Theorem 4.1 and the Gauss–Markov result in linear regression theory (see, for example, [5]). A straightforward proof is included in [12].

Many DOA estimates are asymptotically Gaussian distributed due to the central limit theorem [2],[7]–[11],[13]. The assumption of no correlation between the DOA estimates determined from different subarrays is also likely to be satisfied since the noises in different subarrays are uncorrelated. Thus, the class of estimates considered in Theorem 4.2 is fairly large. The decentralized estimate (4.3) of the DOA vector ω achieves the minimum variance in this class. This estimate, which we name DEMODE-1, can be implemented by means of the following procedure.

DEMODE-1 algorithm

Step 1. Use the MODE procedure of Section 3, along with the data measured in each subarray, to determine the subarray DOA estimates \{ ̂ω_s \}_{s=1}^{L}. At convergence, the MODE algorithms also provide consistent estimates of the inverse covariance matrices \( \{ C_s^{-1} \} \).

Step 2. Using \( \{ ̂ω_s \} \) and \( \{ ̂C_s^{-1} \} \) (s = 1, . . . , L), computed in Step 1, determine the optimal combiner estimate (4.3) of the DOA vector ω.

The computations in Step 1 can be done in parallel, in the local processors associated with each subarray. The results \( \{ ̂ω_s, ̂C_s^{-1} \}_{s=1}^{L} \) of these computations are then transmitted to the central processor which performs Step 2. The computational and communication burdens of this scheme are significantly less than those corresponding to the centralized MODE algorithm.

If the local processing resources are modest but the central processor has significant parallel processing capabilities, then a scheme similar to the one above may still be conceived. However, if none of these conditions is met, then the computation of MODE estimates ̂ω_s must be done serially in the central processor,
and this will increase the computational burden. In this case, where one is forced to do most of the computations in a serial central machine, a computationally more efficient decentralized MODE-based scheme may be devised (as described in the next section).

5. Decentralized MODE with soft local processing requirements

Consider partition (4.1) of the array in \( L \) nonoverlapping subarrays, denote by \( h_s(\omega) \) the function \( h(\omega) \), (3.2), corresponding to subarray \( s \), and let

\[
f(\omega) = \sum_{i=1}^{L} h_s(\omega). \tag{5.1}
\]

Define

\[
\hat{\omega} = \arg \min_{\omega} f(\omega). \tag{5.2}
\]

The statistical properties of \( \hat{\omega} \), (5.2), are established in the following theorem.

**Theorem 5.1.** The estimate \( \hat{\omega} \) given by (5.1), (5.2) is Gaussian distributed in large samples, with mean \( \omega \) and covariance matrix \( C_{L,B} \) defined in (4.4).

**Proof.** See Appendix C.

Thus, the estimates \( \hat{\omega} \) given by (4.3) and (5.2) have the same large-sample statistical properties. In particular, this means that (5.2) is the minimum-variance estimate in a fairly large class of decentralized estimates of the DOA vector \( \omega \) (see Theorem 4.2 and the subsequent discussion). The estimate (5.1), (5.2), which we call DEMODE-2, can be implemented in the following steps.

**DEMODE-2 algorithm**

**Step 1.** Compute the subarray sample covariance matrices \( \{\hat{R}_s\}_{s=1}^{L} \).

**Step 2.** Compute the eigendecompositions of \( \{\hat{R}_s\} \). Use the eigenvectors of \( \{\hat{R}_s\} \) to determine \( n \) by one of the techniques described in [1],[16].

**Step 3.** Determine an initial estimate \( \hat{\omega}^0 \) of \( \omega \) by minimizing the loss function (5.1) with \( \hat{W}_s = I \) \( (s = 1, \ldots, L) \)

\[
f(\omega)_{\hat{W}_s=I} = \sum_{i=1}^{L} h_s(\omega)_{\hat{W}_s=I} = \frac{1}{L} \sum_{s=1}^{L} \sum_{k=1}^{n} a^*_k(\theta_k) G_s \hat{G}_s^* a_s(\theta_k). \tag{5.3}
\]

It follows easily from (5.3) that the elements of \( \hat{\omega}^0 \) are given by the locations of the global minimum points of the following function

\[
y(\theta) = \sum_{i=1}^{L} a^*_i(\theta) \hat{G}_i \hat{G}_i^* a_i(\theta) = \sum_{i=1}^{L} [a^*_i(\theta) a_i(\theta) - a^*_i(\theta) \hat{S}_i \hat{S}_i^* a_i(\theta)]. \tag{5.4}
\]

Thus, \( \hat{\omega}^0 \) coincides with the decentralized MUSIC estimate of [12]!

**Step 4.** Estimate \( \{\hat{W}_s\} \) as

\[
\hat{W}_s = \left[ \{A_s^* \hat{S}_s A_s - \hat{A}_s \hat{S}_s^* A_s\}_{s=1}^{L} \right]^{-1}. \tag{5.5}
\]

**Step 5.** Perform the minimization in (5.2) to determine the optimal decentralized estimate of \( \omega \). This minimization can be conveniently done by means of the following Gauss–Newton algorithm

\[
\hat{\omega}^{k+1} = \hat{\omega}^k - \left\{ \sum_{s=1}^{L} \text{Re}[\hat{H}_s \hat{W}_s^T]_{\omega=\omega^k} \right\}^{-1} \left\{ \sum_{s=1}^{L} \text{Re}[\mu_s]_{\omega=\omega^k} \right\}, \tag{5.6}
\]

where \( \hat{\omega}^0 \) is the initial estimate computed in Step 3, \( \{\hat{W}_s\} \) are the estimates of \( \{\hat{W}_s\} \) computed in Step 4, and \( \hat{H}_s \) and \( \mu_s \) are the quantities defined, respectively, in (3.12) and (3.13), corresponding to subarray \( s \) \( (s = 1, \ldots, L) \).

The computation in Step 1 of the above algorithm is the only one that needs to be done in the local processors. Thus, the DEMODE-2 algorithm has soft local processing requirements. Its computational and communication burdens are significantly less than those associated with the centralized MODE algorithm. Concerning comparison with the DEMODE-1 algorithm, the following comments can be made. In the applications in which the computing resources allow exploiting the parallel structure of DEMODE-1, this algorithm is computationally more efficient than DEMODE-2. Otherwise, DEMODE-2 is computationally more efficient since it requires iteration of one Gauss–Newton recursion (see (5.6)), which is to be contrasted with the \( L \) Gauss–Newton recursions ((3.11) used for each subarray) in the DEMODE-1 algorithm. Note, however, that the difference between the computational burdens of the serial implementations of the two algorithms is not as substantial as it might appear at first sight. Indeed, the evaluation of the gradient and Hessian involved in the single Gauss–Newton recursion of DEMODE-2 requires the same amount of computation as the evaluation of the gradients and Hessians in the \( L \) Gauss–Newton recursions of DEMODE-1. The communication burdens of the two algorithms are fairly similar.

Before ending this section, we comment on another interesting aspect concerning the implementation of the DEMODE algorithms. In the parallel implementation
of DEMODE-1, the initial MUSIC estimates \(\hat{\omega}_0^0\) are computed from each subarray separately. In contrast, in the DEMODE-2 algorithm \(\hat{\omega}_0\) is computed using eigeninformation from all subarrays. One may expect intuitively that \(\hat{\omega}_0\) determined as above is a more accurate estimate of \(\omega\) than any of \(\hat{\omega}_0^0\). A similar comment can be made on the determination of the number of sources, from each subarray output separately or using eigeninformation from all subarrays [16].

We analyzed in [6] the conjecture that \(\hat{\omega}_0\) is more accurate than any of \(\hat{\omega}_0^0\). The covariance matrices of the local MUSIC estimates \(\hat{\omega}_0^0\) and of the decentralized MUSIC estimate \(\hat{\omega}_0\) can be derived, respectively, from the calculations in Appendix A and Appendix C, in which \(\hat{W}\) is set to \(\hat{W} = I\). Between the covariance matrices so obtained there exists no general order relation: each of these matrices can be greater than the other depending on the parameters of the problem under study. Thus, some of the local MUSIC estimates may be more accurate than the decentralized MUSIC estimate [see (6)]. It is worth noting in this context that in the case of the statistically optimal MODE algorithm, the local estimates (whose covariance matrices are \(C_i\)) are always less accurate under the assumptions made, than the decentralized estimate (5.2) (whose covariance matrix is \((\sum_{i=1}^L C_i^{-1})^{-1} \leq C_i\), for all \(i\)).

6. Performance comparison of centralized and decentralized array processing

An important conclusion of the analysis in the previous sections is that the centralized and decentralized MODE algorithms achieve optimal statistical performance in the classes of centralized estimates and, respectively, decentralized DOA estimates determined from nonoverlapping subarrays. Comparing the covariance matrices of the MODE and DEMODE algorithms is therefore equivalent to comparing the best DOA estimation accuracies achievable by centralized and, respectively, decentralized processing of sensor data. This comparison is the subject of the following theorem.

**Theorem 6.1.** The covariance matrices \(C_{CR}\) (3.4) and \(C_{LB}\) (4.4) satisfy the following inequality

\[
C_{LB} \geq C_{CR}
\]

(i.e., the difference matrix \(C_{LB} - C_{CR}\) is positive semidefinite). Furthermore, the equality \(C_{LB} = C_{CR}\) is not possible for \(L > 1\).

**Proof.** The inequality (6.1) follows at once from the fact that \(C_{CR}\) is the CRB on the covariance matrix of any (consistent) DOA estimation method (see Theorem 3.1). An algebraic proof of the theorem can be found in [12, Appendix E]. \(\square\)

Note from the above theorem that the difference matrix \(C_{LB} - C_{CR}\) can never be zero (for \(L \neq 1\)). More quantitative insight into the properties of this difference could not, however, be obtained analytically. In order to obtain such an insight, we resort to numerical evaluations as described in the following example.

**Example 6.1.** Consider a uniform linear array (ULA) composed of \(m = 100\) omni-directional sensors, situated in the wavefield generated by two \((n = 2)\) sources. For a ULA, the vector \(a(\theta)\) has the following form

\[
a(\theta) = \begin{bmatrix} e^{i\theta} & \cdots & e^{i(m-1)\theta} \end{bmatrix}^T.
\]

(Note that \(\theta\) in (6.2) is not effectively the DOA, but it is related to the DOA in a simple way [2],[13]—[15]. In order to simplify the notation, we consider the estimation of \(\theta\) in (6.2) instead of the DOA.) The signals emitted by the sources are assumed to be uncorrelated and of identical power equal to 1. We also assume that \(\sigma = 1\) (thus the signal-to-noise ratio is equal to 0 dB). Finally, we assume that for decentralized processing the array is split into \(L\) subarrays of equal size (thus each subarray contains \(m_s = m/L\) sensors).

The covariance matrices \(C_{CR}\) and \(C_{LB}\), corresponding to the above scenario, have been evaluated for \(L = 2, 4, 10,\) and \(20\), and varying \(\Delta \theta = |\theta_1 - \theta_2|\) (note that in this case the covariance elements are functions of \(\Delta \theta\) only). Figure 1 shows the normalized standard deviations \(\sqrt{[N_{C_{CR}}]}^{1/2} = \sqrt{[N_{C_{LB}}]}^{1/2}\) and \(\sqrt{[N_{C_{LB}}]}^{1/2}\). Note from Figure 1 that the accuracy degradation associated with decentralized processing may be considerable; for large values of \(L\), such as \(L \geq 10\), the DOA estimates obtained by decentralized processing may be quite inefficient statistically. On the other hand, one can see from Figure 1 that the degradation in accuracy corresponding to an increase in \(L\) is not significantly affected by the DOA separation \(\Delta \theta\).

Next note that the curves of DOA standard deviations shown in Figure 1 (for uncorrelated sources) will be only slightly affected if the sources are assumed to be correlated. This is so because the MODE statistical performance is fairly insensitive to the source correlation factor [5]. This property is illustrated in Figure 2, which shows the normalized standard deviation of MODE as a function of \(\Delta \theta\), for \(m = 5\) (the smallest subarray size considered in Figure 1), \(SNR = 0\) dB, and the following values of the source correlation factor \(\rho\): 0, 0.5, and 0.995. It can be seen that the degradation in performance is insignificant indeed, when one passes from uncorrelated to highly correlated sources.

In the case of equally sized subarrays, the decentralized processor has maximum computational advantage over the centralized processor. For an array split in \(L\) subarrays of equal size, the DEMODE algorithms are about \(L^2\) times faster than those of the centralized MODE. However, as shown in Example 6.1, the degradation of accuracy may be significant for equally sized subarrays and it may increase dramatically with increasing \(L\).

The performance degradation associated with decentralized processing may be reduced, at the expense of increasing the computational burden, by using suitably
Example 6.2. Consider the uniform linear array of Example 6.1, made of $m$ sensors, and let $n = 1$. The array is split into two subarrays of lengths $m_1$ and $m_2 = m - m_1$, respectively. Some straightforward calculations show that the covariance matrix $C_{LB}$ corresponding to the above scenario is given by

$$C_{LB} = \frac{6\sigma P N}{\sigma(P) + m_1 + \frac{m_1^2(m_1^2 - 1)}{(\sigma(P) + m_1)(\sigma(P) + m_2)}}.$$  \hspace{1cm} (6.3)

The normalized variance $NC_{LB}$ is plotted in Figure 3 as a function of $m_1$, for $m = 100$ and the following values of $\sigma$ and $P$: $P = 1, \sigma = 10$ (SNR = 0 dB); $P = 1, \sigma = 1$ (SNR = 0 dB); $P = 10, \sigma = 1$ (SNR = 10 dB).

It can be seen that the variance of estimation errors is maximum at $m_1 = m_2$ and that it decreases significantly as $|m_1 - m_2|$ increases. Thus, in this example, the best performance of the decentralized processor is achieved for $m_1 = 2 = n + 1$ (the minimum possible value). However, the computational performance corresponding to this subarray selection is the worst. In fact, the decentralized processor with $m_1 = 2, m_2 = 98$ will behave almost like the centralized one from both the computational and statistical performance points of view.

In practice, the subarray sizes may be constrained by both computational and physical reasons. The constraints may be in the form of lower and upper bounds on $m_s$ (the number of sensors in subarray $s$):

$$m_s \leq m_s \leq \bar{m}_s \quad (s = 1, 2, \ldots, L; \quad L \quad \text{and} \quad m = \sum_{s=1}^{L} m_s \quad \text{are given}).$$  \hspace{1cm} (6.4)

The results in Example 6.2 would suggest that the statistical performance of the decentralized processor might be improved by selecting $m_s = \bar{m}_s$ for as many subarrays as possible. However, this is not necessarily so since $C_{LB}$ may well increase when $m_s$ decreases, as shown in the next example.

Example 6.3. Consider a linear but nonuniform array with the following direction vector

$$a(\theta) = \begin{pmatrix} 1 \\ e^{i\theta} \\ e^{i5\theta} \\ e^{i6\theta} \\ e^{i7\theta} \\ e^{i8\theta} \end{pmatrix}.$$  \hspace{1cm} (6.5)

Assume that there is a single source ($n = 1$), and let $P = \sigma = 1$. Split the array into two subarrays containing the following sensors.

<table>
<thead>
<tr>
<th>subarray 1</th>
<th>subarray 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>sensors #</td>
<td>sensors #</td>
</tr>
<tr>
<td>CASE I</td>
<td>1,2,3</td>
</tr>
<tr>
<td>CASE II</td>
<td>1,2</td>
</tr>
</tbody>
</table>
A simple calculation shows that in the two cases above, the normalized variance $NC_{LB}$ is given by

\begin{align*}
\text{Case I} & \quad NC_{LB} = 1/24 \\
\text{Case II} & \quad NC_{LB} = 3/26
\end{align*}

(regardless of the value of $\theta$). Clearly, the conjecture that $C_{LB}$ would have in the general case, a behavior similar to that exhibited in Example 6.2, is not valid.

Example 6.3 suggests that analytical solutions to the problem of optimally selecting the subarrays so as to minimize $C_{LB}$ subject to the constraints (6.4) may hardly be expected to exist. The subarray selection problem may however be solved in specific cases by using numerical (combinatorial) optimization techniques and the expression for $C_{LB}$ derived in this paper.

7. Concluding remarks

The two methods for decentralized array processing introduced in this paper are statistically efficient in the sense that they provide the most accurate decentralized DOA estimates determined from prescribed nonoverlapping subarrays. The available local and central processing facilities will determine which one of these two methods should be preferred in a specific application. Compared to the similar methods for centralized array processing, the decentralized methods proposed here have greatly reduced computational and communication burdens and, as such, they should be of interest for the increasing number of applications involving arrays with many sensors.

The results of this paper can also be used to study the loss of accuracy associated with decentralized processing. In particular, they can be useful to derive an optimal splitting in subarrays of a large array, i.e., a splitting that minimizes the accuracy degradation under given computational and physical constraints. They can also be useful in designing decentralized processors with prespecified performance.

Appendix A. Proof of (3.4)

The estimate $\hat{\omega}$ (3.1) is consistent. In order to see this, note that, for $N \to \infty$, $h(\omega)$ converges uniformly (w.p.1) to the following function

$$
\tilde{h}(\omega) = \text{tr}[A^*GG^*AW].
$$

Thus, $\hat{\omega}$ converges (w.p.1) to the global minimum of (A.1) [11]. However, it follows from some simple properties of the trace operator (see, for example, [5]) that $\tilde{h}(\omega) \geq 0$ with equality if and only if

$$
A^*G = 0.
$$

The true DOA vector satisfies the equation (A.2) (cf. (2.16)). Moreover, it is the only solution of (A.2) because of assumption A1 (indeed, if another solution existed then, by A1, we would have $n + 1$ linearly independent $m$-vectors orthogonal to an $(m - n)$-dimensional subspace, which is impossible). With this observation the convergence (w.p.1) of $\hat{\omega}$ to $\omega$ is established.

Since $\hat{\omega}$ is a minimum point of $h(\omega)$, it follows that

$$
h'(\hat{\omega}) = 0.
$$

The $k$th component of the gradient vector $h'(\omega)$ is given by

$$
[h'(\omega)]_k = \frac{\partial h(\omega)}{\partial \theta_k} = 2 \text{Re}[d_k^* \hat{\Gamma}^* G^* \hat{W}^{(c)}_k]
$$

where $\hat{W}_k^{(c)}$ denotes the $k$th column of the matrix $\hat{W}$. A Taylor series expansion of (A.3) around the true DOA vector $\omega$ gives

$$
h'(\omega) = h'(\omega) + h''(\omega)(\hat{\omega} - \omega) + \ldots
$$

where $h''(\omega)$ denotes the Hessian matrix. The $(k, p)$-element of this matrix is given by

$$
[h''(\omega)]_{kp} = \frac{\partial h(\omega)}{\partial \theta_k \partial \theta_p} = 2 \text{Re}[d_k^* \hat{\Gamma}^* G^* \hat{W}_k^{(c)} + 2 \text{Re} \left( \frac{d_k^* \hat{\Gamma}^* G^* \hat{W}_k^{(c)}}{\hat{\theta}_k} \right)]
$$

Modulo a trivial permutation of its elements $\{\hat{\theta}_k\}$.
In the following, we use the symbol "≈" to denote an equality in which we have neglected the higher-order terms (i.e., the terms that tend to zero faster than the retained terms, as \( N \) tends to infinity). Using this convention, the Taylor series expansion (A.5) and the fact that \( \hat{G}\hat{G}^* \rightarrow GG^*, A^*\hat{G} \rightarrow 0 \), \( \hat{W} \rightarrow W \), and \( \hat{\omega} \rightarrow \omega \) as \( N \rightarrow \infty \), one can write (for sufficiently large \( N \))

\[
(\hat{\omega} - \omega) \approx -[\text{Re}[H \odot W^T]]^{-1}\text{Re}[\mu]
\]

(A.7)

where the \( k \)th element of \( \mu \) is the following

\[
\mu_k = d_k^r \hat{G}\hat{G}^*AW_k^{(c)}.
\]

(A.8)

Next, note that the middle matrix in (A.8) can be approximated as follows

\[
\hat{G}\hat{G}^*A = \hat{G}\hat{G}^* \hat{G}\hat{G}^*A \approx GG^*\hat{G}\hat{G}^*A
\]

\[
= GG^*(I - \hat{S}\hat{S}^*)A = -GG^*\hat{S}\hat{S}^*A \approx -GG^*\hat{S}\hat{S}^*A.
\]

(A.9)

To proceed with the analysis, let

\[
\hat{\Sigma} = \left(\begin{array}{cc}
\hat{\lambda}_{n+1} & 0 \\
0 & \hat{\lambda}_m
\end{array}\right).
\]

(A.10)

Since \( \hat{\Sigma} \rightarrow \sigma I \) as \( N \rightarrow \infty \) one can write, similarly to (A.9), cf. (2.14),

\[
G^*\hat{R}\hat{S} = G^*(\hat{S}\hat{S}^* + \hat{G}\hat{G}^*)S = (G^*\hat{S}\hat{S}^*)\Lambda + (G^*\hat{G}\hat{G}^*)S
\]

\[
= (G^*\hat{S}\hat{S}^*)\Lambda - \sigma G^*\hat{S}\hat{S}^*S \approx (G^*\hat{S}\hat{S}^*)\Lambda - \sigma G^*\hat{S}\hat{S}^* = (G^*\hat{S}\hat{S}^*)\Lambda
\]

(A.11)

which implies that

\[
G^*\hat{S} \approx G^*\hat{R}\hat{S} \Lambda^{-1} = \frac{1}{N}\sum_{i=1}^{N} G^*(Ax(t) + e(t))y^*(t)S \Lambda^{-1}
\]

\[
= \frac{1}{N}\sum_{i=1}^{N} G^*e(t)y^*(t)S \Lambda^{-1}
\]

(A.12)

Inserting (A.9) and (A.12) in (A.8), we obtain the following convenient expression for \( \mu_k \)

\[
\mu_k \approx -d_k^r GG^* \left[ \frac{1}{N}\sum_{i=1}^{N} e(t)y^*(t) \right] S \Lambda^{-1} S^*AW_k^{(c)}.
\]

(A.13)

The convenience of the above expression lies in the fact that, unlike (A.8), it depends on the original random data explicitly and the dependence on the sample eigenvalues has been eliminated. The equation (A.13) can be written in the following more compact form

\[
\mu_k \approx -\frac{1}{N}\sum_{i=1}^{N} u_k^*e(t)y^*(t)v_k
\]

(A.14)

where

\[
u_k = S \Lambda^{-1} S^*W_k^{(c)}.
\]

(A.16)

It follows from (A.7) that in order to evaluate the asymptotic covariance matrix of \( \hat{\omega} \), we first need to determine the large-sample variances–covariances of \( \{\mu_k\} \).

Using (A.14), we get

\[
E[\mu_k\mu_p] \approx \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} E[u_k^*e(t) \times y^*(t)v_k \times u_p^*e(s) \times y^*(s)v_p]
\]

\[
= \frac{\sigma^2}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \left[ (u_k^*v_k)(u_p^*v_p) + (u_k^*v_p)(u_p^*v_k)\delta_{i,j} \right]
\]

\[
= 0
\]

(A.17)

(where use was made of the fact that \( u_k^*v_p = 0 \) for all \( k \) and \( p \)) and, similarly,

\[
E[\mu_k\mu_p^*] \approx \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} E[u_k^*e(t) \times y^*(t)v_k \times e^*(s)v_p \times y^*(s)v_p^*]
\]

\[
= \frac{\sigma}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \left[ \sigma(u_k^*v_k)(v_p^*A^pA^*v_k) + \sigma^2(u_k^*v_k)(v_p^*v_k)\delta_{i,j} \right]
\]

\[
= \frac{\sigma}{N^2} \sum_{i=1}^{N} (u_k^*v_k)(v_p^*R_kv_k)
\]

\[
= \frac{\sigma}{N^2} \left[ (d_k^rGG^*d_k^r)^{-1} [WA^*S^* \Lambda^{-1} S^*RS^* \Lambda^{-1} S^*AW^*]^p \right]
\]

\[
= \frac{\sigma}{N^2} \left[ [H]^p_k [WA^*S^* \Lambda^{-1} S^*AW]^p \right] = \frac{\sigma}{N^2} [H]^p_k [W]^p_k.
\]

(A.18)

Since

\[
\text{Re}[\mu_k]\text{Re}[\mu_p] = \frac{1}{2}\left\{\text{Re}[\mu_k]\mu_p + \text{Re}[\mu_k^*\mu_p^*] \right\}
\]

it follows from the above calculations of variances–covariances that

\[
E[\text{Re}[\mu_k]\text{Re}[\mu_p]] = \frac{\sigma}{2N} \text{Re}[H \odot W^T]_{kp}
\]

(A.19)

which, along with (A.7), implies that asymptotically,

\[
C_{CR} \triangleq E[(\hat{\omega} - \omega)(\hat{\omega} - \omega)^T] \approx \frac{\sigma}{2N} \left[ \text{Re}[H \odot W^T] \right]^{-1}
\]

(A.20)

and the proof is complete.
Appendix B. Proof of Theorem 4.1

For $\mu = \nu$ the result follows immediately from Theorem 3.1. It remains to prove (4.2) for two different subarrays. The quantities corresponding to the two subarrays are denoted, respectively, by a "bar" and a "tilde" (for convenience of notation). It follows from the proof of Theorem 3.1 in Appendix A that in order to show the uncorrelatedness of the MODE estimates obtained from the two subarrays, it is necessary and sufficient to prove that

$$E\tilde{\mu}_k \tilde{\mu}_p = 0 \quad (B.1)$$

and

$$E\tilde{\mu}_k \tilde{\mu}_p^* = 0. \quad (B.2)$$

However,

$$E\tilde{\mu}_k \tilde{\mu}_p = \frac{1}{N^2} \sum_{t=1}^{N} \sum_{s=1}^{N} E[\tilde{u}_k^* \tilde{\epsilon}(t) \times \tilde{v}_p \tilde{\epsilon}(s) \times \tilde{v}_p^* \tilde{\epsilon}(s) \times \tilde{v}_p \tilde{\epsilon}(s)]$$

$$= \frac{\sigma^2}{N^2} \sum_{t=1}^{N} \sum_{s=1}^{N} (\tilde{u}_k^* \tilde{v}_p) (\tilde{u}_p^* \tilde{v}_p) = 0 \quad (B.3)$$

and

$$E\tilde{\mu}_k \tilde{\mu}_p^* = \frac{1}{N^2} \sum_{t=1}^{N} \sum_{s=1}^{N} E[\tilde{u}_k^* \tilde{\epsilon}(t) \times \tilde{v}_p^* \tilde{\epsilon}(s) \times \tilde{v}_p \tilde{\epsilon}(s) \times \tilde{v}_p^* \tilde{\epsilon}(s)]$$

$$= \frac{\sigma^2}{N^2} \sum_{t=1}^{N} \sum_{s=1}^{N} (\tilde{u}_k \tilde{v}_p^*) (\tilde{u}_p \tilde{v}_p) = 0. \quad (B.4)$$

and the proof is ended.

Appendix C. Proof of Theorem 5.1

It follows from the defining relation (5.1) of $f(\omega)$ that

$$f'(\omega) = \sum_{l=1}^{L} h'_l(\omega) \quad (C.1)$$

$$f''(\omega) = \sum_{l=1}^{L} h''_l(\omega). \quad (C.2)$$

Using this fact and paralleling the calculations (A.3)–(A.7) in Appendix A, one can show that

$$(\hat{\omega} - \omega) \simeq \left[ \frac{\sigma}{2N} \sum_{s=1}^{L} C_s^{-1} \right]^{-1} \left\{ \text{Re} \left[ \sum_{l=1}^{L} \mu_s \right] \right\}. \quad (C.3)$$

Since the random variables $\{\text{Re} \mu_s\}_{s=1}^{L}$ are asymptotically Gaussian distributed (see Theorem 3.1 and Appendix A) and uncorrelated (see Appendix B), it readily follows from (C.3) that $\hat{\omega}$ is Gaussian distributed in large samples, with mean $\omega$ and the following covariance matrix

$$E(\hat{\omega} - \omega)(\hat{\omega} - \omega)^T = \left[ \frac{\sigma}{2N} \sum_{s=1}^{L} C_s^{-1} \right]^{-1} \times \left\{ \sum_{s=1}^{L} E[\text{Re} \mu_s \text{Re} \mu_s^*] \right\} \left[ \frac{\sigma}{2N} \sum_{s=1}^{L} C_s^{-1} \right]^{-1} \quad (C.4)$$

However (see equation (A.19) in Appendix A)

$$E[\text{Re} \mu_s \text{Re} \mu_s^*] = \left( \frac{\sigma}{2N} \right)^2 C_s^{-1} \quad (C.5)$$

Inserting (C.5) in (C.4), we get

$$E(\hat{\omega} - \omega)(\hat{\omega} - \omega)^T = \left[ \sum_{s=1}^{L} C_s^{-1} \right]^{-1} \quad = C_{LB}$$

and the proof is ended.

Acknowledgments

We are very grateful to Ms. Ylva Johansson for her excellent assistance in preparing this paper.

References


