Array Shape Calibration Using Sources in Unknown Locations—A Maximum Likelihood Approach

ANTHONY J. WEISS, SENIOR MEMBER, IEEE, AND BENJAMIN FRIEDLANDER, FELLOW, IEEE

Abstract—Sensor location uncertainty can severely degrade the performance of direction finding systems. An iterative maximum likelihood method for simultaneously estimating directions of arrival (DOA) and sensor locations is developed to alleviate this problem. The case of nondisjoint sources, i.e., sources observed in the same frequency cell and at the same time, is emphasized. The algorithm converges to the global maximum of the likelihood function if the initial conditions are sufficiently good. Numerical examples are presented, illustrating the performance of the proposed technique.

I. INTRODUCTION

The localization of radiating sources by a passive array of sensors is a problem of considerable importance in a variety of fields ranging from radar, sonar, oceanography, and seismology to radio astronomy. This problem has, therefore, received considerable attention in the literature, resulting in many different estimation schemes.

Source localization techniques based on eigenstructure methods have been discussed extensively in the literature since the beginning of this decade. Computer simulations and a relatively limited number of experimental systems have demonstrated that in certain cases, these techniques have superior performance compared to conventional direction finding techniques.

In spite of the potential advantages of eigenstructure methods, their application to real systems has been very limited. One of the main reasons for this situation is the practical difficulties associated with calibrating the data collection system. Eigenstructure-based direction finding techniques such as MUSIC [7] require precise knowledge of the signals received by the sensor array from a standard source located at any direction. The collection of the received signal vectors for all possible directions is often called the array manifold. The performance of the eigenstructure-based system depends strongly on the accuracy of this array manifold.

Direction finding techniques based on the maximum likelihood (ML) approach have been introduced by various authors. A general ML technique which applies to arbitrary array configurations, signal bandwidth, and number of sources is the EM algorithm [10] and the iterative technique presented in [1].

In this paper, we are interested in source localization in the presence of sensor location uncertainty. Such uncertainty occurs, for example, in towed arrays or in tactical systems which require that the array be dismantled and reassembled in the field. Rockah and Schultheiss [3]–[4] have studied this problem in detail, using the Cramer–Rao lower bound for the DOA estimates in the presence of sensor location uncertainty. They have shown that finding the directions to the sources is possible if one sensor location and the direction to a second sensor are known. Moreover, they presented an algorithm for self-calibration which is based on observing disjoint sources (sources which can be separated either in the time domain or in the frequency domain). In this paper, we emphasize the case of nondisjoint sources, using an iterative maximum likelihood procedure.

Lo and Marple [2] discussed a calibration technique which requires calibrating sources whose directions are known (at least two sources are required). Therefore, their technique is not a true self-calibrating technique.

In a companion paper [5], we address the self-calibration problem for an array with imprecisely known sensor gains and phases. The combination of the techniques presented here and in [5] is expected to be useful for alleviating some of the practical problems associated with eigenstructure-based direction finding methods.

The outline of the paper is as follows. In Section II, we formulate the problem and discuss some necessary conditions for the existence of a solution. Section III describes an algorithm for simultaneous estimation of the source DOA's and the unknown sensor locations. In Section IV, we present selected numerical examples which illustrate the performance of the proposed algorithm.

II. PROBLEM FORMULATION

Consider \( N \) radiating sources observed by an array of \( M \) sensors. The signal at the output of the \( m \)th sensor can be described by

\[
    r_m(t) = \sum_{n=1}^{N} s_n(t - \tau_{mn}) + v_m(t);
    \quad m = 1, 2, \ldots, M; -T/2 \leq t \leq T/2
\]

(1)

where \( \{ s_n(t) \}_{n=1}^{N} \) are the radiated signals, \( \{ v_m(t) \}_{m=1}^{M} \) are sample waveforms from additive noise processes, and \( T \) is the observation interval. The parameters \( \{ \tau_{mn} \} \) are delays associated with the signal propagation time from...
the $n$th source to the $m$th sensor. These parameters are of interest since they contain information about the source location relative to the array.

A convenient separation of the parameters to be estimated is obtained by representing the signals using Fourier coefficients defined by

\[
R_m(t) = \frac{1}{\sqrt{T}} \int_{-T/2}^{T/2} r_n(t) e^{-j\omega t} dt
\]

where $\omega_l = (2\pi/T)(l_1 + l_2)$, $l = 1, 2, \cdots, L$ and $l_1$ is a constant. In principle, the number of coefficients required to capture all the signal information is infinite. However, if we consider signals with energy concentrated in a finite spectral band, we can use a finite number of coefficients. Moreover, in this work, we consider only narrow-band signals with energy concentrated around $\omega_0$, with a bandwidth that is small compared to $2\pi/T$, and hence $L = 1$. Taking the Fourier coefficients of (1) and suppressing the dependence on $\omega_0$, we obtain

\[
R_m(j) = \sum_{n=1}^{N} e^{-j\omega_0 t_n} S_n(j) + V_n(j)
\]

where $j$ denotes the (time) index of the different samples, $S_n(j)$ and $V_n(j)$ are the Fourier coefficients of $s_n(t)$ and $v_n(t)$, respectively. Equation (3) may be expressed using vector notation as follows:

\[
R(j) = AS(j) + V(j); \quad j = 1, 2, \cdots, J
\]

where

\[
R(j) = [R_1(j), R_2(j), \cdots, R_L(j)]^T
\]

\[
S(j) = [S_1(j), S_2(j), \cdots, S_n(j)]^T
\]

\[
V(j) = [V_1(j), V_2(j), \cdots, V_M(j)]^T
\]

\[
A_{mn} = e^{-j\omega_0 t_n};
\]

$m = 1, 2, \cdots, M$, $n = 1, 2, \cdots, N$.

In this paper, we assume that the sensors and sources are coplanar and the sources are far enough from the observing array so that the signal wavefronts are effectively planar over the array. It is easy to verify that the delays $\tau_{mn}$ are given by

\[
\tau_{mn} = -d_{mn}/c
\]

\[
d_{mn} = x_m \sin \theta_n + y_m \cos \theta_n
\]

where $c$ is the propagation velocity, $d_{mn}$ is the distance from sensor $m$ to sensor 1 (reference sensor) in the direction of the $n$th source, ($x_m, y_m$) are the coordinates of sensor $m$, $\theta_n$ is the DOA of the $n$th source relative to the $y$ axis, and the origin of the Cartesian coordinate system coincides with sensor number 1; see Fig. 1.

Under the assumption that the number of sources is known, the least squares estimate of $\{\theta_n\}_{n=1}^{N}$ and $\{(x_m, y_m)\}_{m=1}^{M}$ is given by

\[
\hat{\theta}_n, \{\hat{x}_m, \hat{y}_m\} = \arg \min Q
\]

\[
Q = \sum_{j=1}^{J} \| R(j) - AS(j) \|^2
\]

where $\| \cdot \|$ denotes the Euclidean norm. Equation (6) also represents the maximum likelihood estimates under the assumption that the noise vectors $\{V(j)\}$ are i.i.d. zero-mean Gaussian with covariance $\sigma^2 I$.

It is clear from the discussion in [3] and [4] that this problem has a unique solution only if: the array is nonlinear, at least three spatially disjoint sources are present, and the location of one sensor and the direction of another sensor are known. This is the case that we treat here. However, if, for example, the direction from sensor 1 to one of the other sensors is not known, the technique described in the next section can still be applied. In this case, we will achieve an array shape calibration, but will not be able to estimate correctly the array orientation. As a result, the DOA's will be estimated up to an unknown rotation angle. In other words, the differences between the DOA's will be estimated accurately, but not their absolute values.

### III. The Estimation Procedure

The minimization required in (6) is not trivial since the vectors $S(j)$ and the matrix $A$ are not known to the observer. However, whenever $A$ is known, $Q$ is minimized by choosing

\[
\hat{S}(j) = (A^H A)^{-1} A^H R(j)
\]

as the estimates of $S(j)$ for $j = 1, 2, \cdots, J$. Relation (8) enables us to update the estimates $\hat{S}(j)$ whenever we have a new estimate for $A$.

The proposed algorithm consists of two steps. In the first step we estimate the DOA's using, at first, the nominal sensor locations, and later using estimates provided by the second step. During the second step, we estimate sensor locations based on the DOA's provided by the first step. The algorithm iterates between the two steps until convergence is achieved.

**The First Step**

During the first step, the algorithm performs successive minimization operations on each column of $A$, holding all the rest of the columns and the associated components of
\( S(j) \) fixed. For example, suppose that we want to perform a minimization with respect to the \( k \)th column vector; then \( Q \) can be rewritten as

\[
Q = \sum_{j=1}^{J} \left\| R^k(j) - a(\theta_k) S_k(j) \right\|^2
\]  

(9)

where \( a(\theta_k) \) is the \( k \)th column of \( A \), \( S_k(j) \) is the \( k \)th component of \( S(j) \), and \( R^k(j) \) is given by

\[
R^k(j) = R(j) - AS^k(j)
\]

(10)

where \( S^k(j) \) is simply \( S(j) \) with the \( k \)th component replaced by zero.

The minimization of (9) with respect to \( a(\theta_k) \) using (8) with \( A \) replaced by \( a(\theta_k) \) is given by

\[
\hat{\theta}_k = \arg \min_{\theta_k} \frac{1}{M} \sum_{j=1}^{J} \left\| R^k(j) - \frac{1}{M} a(\theta_k)^H R^k(j) \right\|^2.
\]

(11)

which is equivalent to

\[
\hat{\theta}_k = \arg \max_{\theta_k} \frac{1}{M} \sum_{j=1}^{J} \left\| R^k(j) \right\|^2.
\]

(12)

The maximization defined by (12) can be performed by a search over the space of \( \theta_k \). This first step of the estimation procedure can thus be summarized as follows:

1) Construct the matrix \( A = A^{0} \) using, initially, the nominal sensor locations, and at a later stage, the last estimate of sensor locations. For DOA's, use an initial guess at first, and at a later stage, the last DOA estimates. Set \( k = 1 \).

2) Compute \( S(j) \) according to (8).

3) Compute \( R^k(j) \) according to (10).

4) Find \( \hat{\theta}_k \) according to (12), and compute the matrix \( A \).

5) Check convergence of \( Q \) [given by (7)].

If no, set \( k = k + 1 \); go back to 2).

If yes, proceed to the second step.

This step of the algorithm is an implementation of the maximum likelihood DOA estimator for the case of known sensor locations; see, e.g., [1].

**The Second Step**

In this step, we use the DOA estimates provided by the first phase to estimate the sensor locations. Suppose that the last estimate of sensor locations (or the nominal sensor locations) is given by

\[
x_0 = \left[ x_{00}, x_{01}, \ldots, x_{0M} \right]^T
\]

\[
y_0 = \left[ y_{00}, y_{01}, \ldots, y_{0M} \right]^T.
\]

(13)

We want to minimize \( Q \) (7) with respect to \( \{x_m\}_{m=1}^{M} \) and \( \{y_m\}_{m=1}^{M} \). Now note that

\[
Q = \sum_{m=1}^{M} \sum_{j=1}^{J} \left| R_m(j) - \sum_{n=1}^{N} A_{m,n} S_n(j) \right|^2 = \sum_{m=1}^{M} Q_m
\]

(14)

where

\[
Q_m = \sum_{j=1}^{J} \left| R_m(j) - \sum_{n=1}^{N} A_{m,n} S_n(j) \right|^2.
\]

(15)

Note that each \( Q_m \) can be minimized separately with respect to \( (x_m, y_m) \) since only \( Q_m \) is a function of \( (x_m, y_m) \).

The minimization of \( Q_m \) with respect to \( x_m \) and \( y_m \) is a nonlinear problem that may be accomplished in several ways. The straightforward (and not so attractive) way is to perform a simple two-dimensional search. An alternative way is to perform a few Newton iterations. We use a closed form solution that relies on the assumption that the sensors are rather close to their nominal locations. We first expand \( Q_m \) as follows:

\[
Q_m = \sum_{j=1}^{J} \left| R_m(j) - \sum_{n=1}^{N} A_{m,n} S_n(j) \right|^2
\]

\[
= \exp \left\{ \int \frac{\omega_0}{c} \left[ \Delta x_m \sin \hat{\theta}_n + \Delta y_m \cos \hat{\theta}_n \right] \right\}^2
\]

\[
= 1 + \frac{\omega_0}{c} \Delta x_m \sin \hat{\theta}_n + \frac{\omega_0}{c} \Delta y_m \cos \hat{\theta}_n.
\]

(16)

(17)

Under the assumption that

\[
\Delta x_m \sin \hat{\theta}_n + \Delta y_m \cos \hat{\theta}_n \ll \frac{\lambda}{2}
\]

where \( \lambda \) is the signal wavelength, we have

\[
\exp \left\{ \int \frac{\omega_0}{c} \left[ \Delta x_m \sin \hat{\theta}_n + \Delta y_m \cos \hat{\theta}_n \right] \right\}
\]

\[\equiv 1 + \frac{\omega_0}{c} \Delta x_m \sin \hat{\theta}_n + \frac{\omega_0}{c} \Delta y_m \cos \hat{\theta}_n.
\]

(18)

This assumption of small location errors is a practical one in many situations. For example, in the case of a towed array, the sensor locations are measured by instrumentation to a certain accuracy, and the residual errors are fairly small. In the case of antenna arrays, the element locations are fixed up to small deviations due to mechanical or thermal effects. If the small error assumption does not hold, the technique proposed here will still work. However, the following algorithm needs to be modified.

Now \( Q_m \) can be expressed as

\[
Q_m = \sum_{j=1}^{J} \left| R_m(j) - P_m(j) \Delta x_m - K_m(j) \Delta y_m \right|^2
\]

(19)

where

\[
R_m(j) = R_m(j) - \sum_{n=1}^{N} A_{m,n} S_n(j)
\]

\[
P_m(j) = j(\omega_0/c) \sum_{n=1}^{N} A_{m,n} S_n(j) \sin \hat{\theta}_n
\]

\[
K_m(j) = j(\omega_0/c) \sum_{n=1}^{N} A_{m,n} S_n(j) \cos \hat{\theta}_n.
\]

(20)
Equation (19) can be rewritten as

\[ Q_m = \left\| \hat{R}_m - B_m \left( \frac{\Delta x_m}{\Delta y_m} \right) \right\|^2 \]  

(21)

where

\[ \hat{R}_m(j) = [\hat{R}_m(1), \hat{R}_m(2), \cdots, \hat{R}_m(J)]^T \]

\[ B_m = \begin{bmatrix}
    P_m(1) & K_m(1) \\
    P_m(2) & K_m(2) \\
    \vdots & \vdots \\
    P_m(J) & K_m(J)
\end{bmatrix} \]

(22)

Obviously, \( Q_m \) in (21) is minimized by the selection

\[ \left( \frac{\Delta \hat{x}_m}{\Delta \hat{y}_m} \right) = \Re \left\{ B_m^H B_m \right\}^{-1} \Re \left\{ B_m^H \hat{R}_m \right\} \]

(23)

which is a closed form solution for real \( \Delta x_m \) and \( \Delta y_m \) that minimizes \( Q_m \), as required.

The second step of the algorithm can thus be summarized as follows.

1. Using \( \{\theta_n\}_{n=1}^N \) provided by the first phase and the last estimates of \( \{x_m, y_m\}_{m=1}^M \), construct the matrix \( A \).
   Set \( m = 1 \).
2. Estimate \( S(j) \) according to (8).
3. Estimate \( \{x_m, y_m\} \) according to (23).
4. Update \( A \).
   Set \( m = m + 1 \). If \( m \leq M \), go back to 2).
   If \( m > M \), continue.
5. Check convergence of \( Q \) [given by (7)].
   If no, set \( m = 1 \) and go back to 2).
   If yes, go to the first step.

Note that the likelihood function (cost function) defined by (7) depends only on the variables \( \{x_m, y_m\}_{m=1}^M \) and \( \{\theta_n\}_{n=1}^N \). The unknown \( S(j) \) can be expressed in terms of these parameters as shown by (8). Recall that in the first step of the algorithm, we hold all the location parameters \( \{x_m, y_m\}_{m=1}^M \) fixed and minimize \( Q \) with respect to the DOA’s \( \{\theta_n\}_{n=1}^N \). The value of \( Q \) will decrease (or remain unchanged) during this step since the updates of \( \theta_n \) and \( S(j) \) involve minimization operations [see 1) and 4) in the summary of the first step].

In the second step of the algorithm, we fix the estimates of \( \{\theta_n\}_{n=1}^N \) and minimize over the parameters \( \{x_m, y_m\}_{m=1}^M \). The value of the cost function will again be reduced (or will stay the same) since the updates of \( \{x_m, y_m\} \) and of \( S(j) \) involve minimization operations [see 2) and 4) in the summary of the second step]. If \( Q_k \) is the value of \( Q \) at iteration \( k \), we must have \( Q_1 \geq Q_2 \geq \cdots \geq Q_N > 0 \). This means that the sequence \( Q_k \) is a convergent sequence. In other words, the cost function will converge to some positive constant value. The convergence point of the algorithm may be either a global or a local minimum, depending on the initial conditions of the algorithm. If the initial values of \( \{x_m, y_m\}_{m=1}^M \) and \( \{\theta_n\}_{n=1}^N \) are sufficiently close to the true sensor locations, convergence to the global minimum will occur.

IV. NUMERICAL EXAMPLES

We have evaluated the performance of this algorithm by a large number of computer simulations. As is characteristic of iterative maximum likelihood procedures, algorithm initialization is a key issue. Using the MUSIC algorithm to provide initial conditions, we found that the ML algorithm was able to significantly reduce the initial estimation errors in all cases where the MUSIC estimates were in the general neighborhood of the correct DOA’s. Given completely erroneous initial conditions, it is possible to find cases where the proposed iterative procedure fails to converge to the global minimum, and therefore produces incorrect results. This, again, is typical of maximum likelihood procedures which do not perform a global search (which is computationally prohibitive in most situations).

In this section, we present a typical numerical example which illustrates the behavior of the algorithm. This example is motivated by problems encountered in tactical HF communication arrays.

Consider a uniform circular array of six sensors separated by half a wavelength of the actual narrow-band source signals, as shown in Fig. 2. Three equal-power narrow-band sources are located in the far field of the array at directions \( \theta_1 = -35^\circ \), \( \theta_2 = 0^\circ \), \( \theta_3 = 35^\circ \). Additive uncorrelated sensor noise is injected with an SNR of 30 dB. Two-hundred snapshots of array data are accumulated prior to the application of the algorithm. The actual locations of the sensors are perturbed by up to 34 percent of \( \lambda/2 \) as shown in Fig. 3. The location of sensor 1 and the direction to sensor 2 are assumed to be known. The MUSIC algorithm was applied first in order to obtain initial guess of the DOA’s. The result was \( \hat{\theta}_1 = -34.8^\circ \), \( \hat{\theta}_2 = 4.1^\circ \), and \( \hat{\theta}_3 = 41.6^\circ \). Then the proposed algorithm was applied. The algorithm converged to within 0.2° in 30 iterations as shown in Table I. Fig. 4 shows the sensor location error as a function of the algorithm iterations. The vertical axis is the average location error in percent of \( \lambda/2 \). Before the first iteration, the average location error is 18 percent, and after the last iteration, the error is less than 0.3 percent.

The average location error is obtained by computing the distances between the current sensor location estimates and the true sensor locations. These distances are averaged over all sensors. All distances are measured in half wavelength units. For example, the average location error of 18 percent means that the distance between the true sensor location and the estimated location, averaged over all the sensors, was 0.18\( \lambda/2 \) where \( \lambda \) is the wavelength.

Fig. 5 shows the norm of the DOA errors in degrees as a function of iterations. It is observed that the first iteration increased the DOA error, but reduced the cost function considerably. Note that the norm of the DOA errors was reduced from 8° to 0.22° in 30 iterations. Fig. 6 presents the cost function \( Q \) defined in (7) as a function of the number of iterations. As expected, the cost is reduced
at each iteration. Finally, Fig. 7 shows the spatial spectrum of the MUSIC algorithm for two cases. The first case is the result obtained by the MUSIC algorithm prior to the application of the proposed algorithm. This estimate was used to initialize the algorithm. The second case is the result obtained by using the sensor location estimates obtained from the proposed algorithm. Note the dramatic improvement in performance. As expected, the MUSIC DOA estimates obtained in the second case coincide with the DOA estimates of the proposed algorithm.

Monte Carlo Experiments

To demonstrate the statistical efficiency of the proposed procedure under different signal-to-noise conditions, we performed the following computer simulations.
The setup described in the previous section was used. The sensors and sources were kept at the same locations. We collected 60 sets of random data: 20 data sets for each of the SNR values of 30, 20, and 10 dB. Each of these data sets contained 200 snapshots. The proposed algorithm was applied to each set. The algorithm termination criterion was selected to be a 0.1 percent reduction of the cost function value in a single iteration.

The algorithm output was used to obtain an estimate of the standard deviation and bias of the directions of arrival and of the sensor coordinates. These values are plotted in Figs. 8–11 together with the 0.99 confidence intervals and the theoretical limit provided by the Cramer–Rao lower bound (CRLB). In the Appendix, we develop simple formulas for an exact numerical evaluation of the bound.

Fig. 8 depicts the CRLB and the standard deviation for the DOA estimates of the source at 35°. The bias for these estimates is plotted in Fig. 9. The corresponding plots for the other sources are quite similar and have, therefore, been omitted. Fig. 10 depicts the CRLB and the standard deviation for the X coordinate estimates of the sixth sensor. The bias for these estimates is plotted in Fig. 11. These plots are representative of all the sensors and for the Y coordinates as well.

Examination of these plots indicates that the standard deviation of the DOA estimates achieves the theoretical bound for a range of signal-to-noise ratios. The bias of the estimates increases as the signal-to-noise ratio decreases.
V. CONCLUSIONS

We presented an iterative algorithm for obtaining maximum likelihood estimates of the DOA's of wavefronts impinging on an array, and at the same time estimating the sensor locations. Perhaps the most distinctive feature of the algorithm is its ability to locate the sensors accurately, without deploying calibration sources at known locations. The algorithm does not require disjoint sources or sources that can be turned on and off as proposed in [2]-[4]. The proposed technique can be used to improve the initial DOA estimates provided by an algorithm which does not take into account sensor location uncertainties. Although the derivations and numerical examples were performed for narrow-band sources, the method can be modified in a straightforward manner to handle wide-band sources.

APPENDIX

THE CRAMÉR–RAO LOWER BOUND FOR JOINT ESTIMATION OF DOA'S AND SENSOR LOCATIONS

Consider the vector of unknown nonrandom parameters

\[
\Theta = \left[ \gamma_1, \gamma_2, \ldots, \gamma_N, x_1, x_2, \ldots, x_M, y_1, y_2, \ldots, y_M \right]
\]  

where \( \gamma_i \) denotes the direction of arrival of the \( i \)th source and \( x_j, y_j \) represent the Cartesian coordinates of the \( j \)th sensor. It is well known that when \( \Theta \) is imbedded in the covariance of a zero-mean Gaussian observation vector, then the components of the Fisher information matrix (FIM) are given by

\[
J_{mn} = -E \left( \frac{\partial^2 L}{\partial \theta_m \partial \theta_n} \right)
= K \cdot \text{tr} \left( R^{-1} \cdot \frac{\partial R}{\partial \theta_m} \cdot R^{-1} \cdot \frac{\partial R}{\partial \theta_n} \right)
\]  

(A.1)

Here, \( L \) stands for the logarithm of the probability distribution function associated with \( K \) statistically independent observations, and \( R \) denotes the real covariance matrix of these data. In the case considered here, we have

\[
R = AR \cdot A^H + \sigma^2 I
\]  

(A.3)

where \( R \) is the signal covariance matrix, \( \sigma^2 I \) is the noise covariance matrix, and \( A \) is the signal directions matrix as defined by (4). To simplify the derivation, we assume that \( \sigma^2 \) and \( R \) are known. Dividing both sides by \( \sigma^2 \), we obtain

\[
\frac{1}{\sigma^2} R = APA^H + I
\]  

(A.4)

where

\[
P \triangleq \frac{1}{\sigma^2} R_i.
\]  

(A.5)

It is easy to verify that the FIM does not change if \( R \) is replaced with \( \hat{R} \).

In the sequel, we use the following notation to simplify the formulas:

\[
W \triangleq A^H A
\]  

(A.6)

and

\[
Q \triangleq (P^{-1} + W)^{-1}.
\]  

(A.7)

We also use the relation

\[
R^{-1} = I - AQA^H
\]  

(A.8)

which implies that

\[
P W Q = Q W P = P - Q.
\]  

(A.9)

From (A.1) and (A.2), we observe that the FIM can be partitioned into nine submatrices as follows:

\[
J = \begin{bmatrix}
J_{\gamma \gamma} & J_{\gamma x} & J_{\gamma y} \\
J_{x \gamma} & J_{xx} & J_{xy} \\
J_{y \gamma} & J_{yx} & J_{yy}
\end{bmatrix}
\]  

(A.10)

where \( J_{\gamma \gamma} \) is the submatrix associated with derivatives with respect to the DOA’s \( \gamma_i \); and \( J_{xx} \) and \( J_{yy} \) are submatrices associated with derivatives with respect to \( x_i \) and \( y_i \), respectively. The off-diagonal submatrices are associated with the various cross terms.

The following is a detailed derivation of each of the above submatrices.

Derivation of the Matrix \( J_{\gamma \gamma} \):

In the ensuing development, we make use of the notational device

\[
\dot{A}_{\gamma j} \triangleq \frac{\partial A}{\partial \gamma_j}
\]  

(A.11)

for the partial derivative of the matrix \( A \) with respect to the DOA \( \gamma_j \) of the \( j \)th source. Using (A.11), we first write the partial derivative of the covariance matrix with respect
to the jth DOA as
\begin{equation}
\frac{\partial R}{\partial \gamma_j} = \hat{A}_{\gamma_j} P A^H + A P \hat{A}_{\gamma_j}^H, \tag{A.12}
\end{equation}
Substituting in (A.2) and noting that
\begin{equation}
\text{tr} (A^H) = \text{conj} \{ \text{tr} (A) \} \tag{A.13}
\end{equation}
we obtain
\begin{equation}
J_{\gamma_j} = 2 \text{Re} \left\{ \text{tr} \left\{ \hat{A}_{\gamma_j} P A^H R^{-1} \right\} \hat{A}_{\gamma_j} P A^H R^{-1} \right\} + \text{tr} \left\{ \hat{A}_{\gamma_j} P A^H R^{-1} \hat{A}_{\gamma_j} P A^H R^{-1} \right\}. \tag{A.14}
\end{equation}
Observe that
\begin{equation}
\hat{A}_{\gamma_j} = \hat{A}_{\gamma_j} e_i e_i^T \tag{A.15}
\end{equation}
where the unit vector $e_i$ is the jth column vector of the $M \times M$ identity matrix and $\hat{A}_{\gamma_j}$ is the matrix of derivatives given by
\begin{equation}
\hat{A}_{\gamma_j} = \sum_{n=1}^{N} \frac{\partial A}{\partial \gamma_j} = \left[ \dot{a}(\gamma_1), a(\gamma_2), \cdots, \dot{a}(\gamma_N) \right]. \tag{A.16}
\end{equation}
Using (A.15), (A.14) becomes
\begin{equation}
J_{\gamma_j} = 2 \text{Re} \left\{ \text{tr} \left\{ \hat{A}_{\gamma_j} e_i e_i^T P A^H R^{-1} A e_i e_i^T P A^H R^{-1} \right\} + \text{tr} \left\{ \hat{A}_{\gamma_j} e_i e_i^T P A^H R^{-1} A e_i e_i^T P A^H R^{-1} \right\} \right\}
= 2 \text{Re} \left\{ e_i^T P A^H R^{-1} A e_i e_i^T P A^H R^{-1} \hat{A}_{\gamma_j} e_i \right\}
+ e_i^T P A^H R^{-1} \hat{A}_{\gamma_j} e_i e_i^T P A^H R^{-1} A e_i \right\}. \tag{A.17}
\end{equation}
Hence,
\begin{equation}
J_{\gamma_j} = 2 \text{Re} \left\{ (P A^H R^{-1} A) \times (\hat{A}_{\gamma_j} e_i e_i^T )^T \right\}
+ (P A^H R^{-1} A) \times (\hat{A}_{\gamma_j} e_i e_i^T )^T \right\} \tag{A.18}
\end{equation}
where $J_{\gamma_j}$ is the submatrix of the FIM associated with the DOA derivatives and $\times$ denotes the Hadamard product of 2 matrices, defined by
\begin{equation}
(A \times B)_{ij} = A_{ij} B_{ij}. \tag{A.19}
\end{equation}
Equation (A.18) may be further simplified using (A.9) as follows:
\begin{equation}
J_{\gamma_j} = 2 \text{Re} \left\{ (P - Q) \times (\hat{A}_{\gamma_j} e_i e_i^T )^T \right\}
+ (Q A^H A_{\gamma_j}) \times (Q A^H A_{\gamma_j})^T \right\}, \tag{A.20}
\end{equation}
which is the desired result.

**Derivation of the Matrices $J_{xx}$ and $J_{yy}$**

The derivation of $J_{xx}$ and $J_{yy}$ follows the same steps as above. First, observe that
\begin{equation}
\frac{\partial R}{\partial x_i} = \hat{A}_{x_i} P A^H + A P \hat{A}_{x_i}^H, \tag{A.21}
\end{equation}
Substituting in (A.2), we obtain
\begin{equation}
J_{x_i x_i} = 2 \text{Re} \left\{ \text{tr} \left\{ \hat{A}_{x_i} P A^H R^{-1} \right\} \hat{A}_{x_i} P A^H R^{-1} \right\} + \text{tr} \left\{ \hat{A}_{x_i} P A^H R^{-1} \hat{A}_{x_i} P A^H R^{-1} \right\}. \tag{A.22}
\end{equation}
Observe that
\begin{equation}
\hat{A}_{x_i} = e_i e_i^T \hat{A}, \tag{A.23}
\end{equation}
where $\hat{A}$ is the matrix of derivatives given by
\begin{equation}
\hat{A} \triangleq \sum_{m=1}^{M} \frac{\partial A}{\partial x_m} \frac{\partial A}{\partial x_m}, \tag{A.24}
\end{equation}
Using (A.23), (A.22) becomes
\begin{equation}
J_{x_i x_i} = 2 \text{Re} \left\{ \text{tr} \left\{ e_i e_i^T \hat{A} P A^H R^{-1} e_i e_i^T P A^H R^{-1} \right\} + \text{tr} \left\{ e_i e_i^T \hat{A} P A^H R^{-1} e_i e_i^T P A^H R^{-1} \right\} \right\}
= 2 \text{Re} \left\{ \text{tr} \left\{ e_i^T \hat{A} P A^H R^{-1} e_i e_i^T P A^H R^{-1} e_i \right\}
+ \text{tr} \left\{ e_i^T \hat{A} P A^H R^{-1} e_i e_i^T P A^H R^{-1} e_i \right\} \right\}. \tag{A.25}
\end{equation}
This leads to
\begin{equation}
J_{xx} = 2 \text{Re} \left\{ (\hat{A} P A^H R^{-1} \times (\hat{A} P A^H R^{-1})^T
+ (\hat{A} P A^H R^{-1} \hat{A} P A^H R^{-1}) \times (R^{-1} \times R^{-1}) \right\} \tag{A.26}
\end{equation}
or, equivalently,
\begin{equation}
J_{xx} = 2 \text{Re} \left\{ (\hat{A} P A^H R^{-1} \times (\hat{A} P A^H R^{-1})^T
+ (\hat{A} P A^H R^{-1} \hat{A} P A^H R^{-1}) \times (R^{-1} \times R^{-1}) \right\}. \tag{A.27}
\end{equation}
The analogous result for $J_{yy}$ is
\begin{equation}
J_{yy} = 2 \text{Re} \left\{ (\hat{A} P A^H R^{-1} \times (\hat{A} P A^H R^{-1})^T
+ (\hat{A} P A^H R^{-1} \hat{A} P A^H R^{-1}) \times (R^{-1} \times R^{-1}) \right\}. \tag{A.28}
\end{equation}

**The Cross-Terms Matrices**

Using exactly the same procedure, it is straightforward to verify that the cross-terms matrices are given by
\begin{equation}
J_{xy} = 2 \text{Re} \left\{ (\hat{A} P A^H R^{-1} \times (\hat{A} P A^H R^{-1})^T
+ (\hat{A} P A^H R^{-1} \hat{A} P A^H R^{-1}) \times (R^{-1} \times R^{-1}) \right\} \tag{A.29}
\end{equation}
\begin{equation}
J_{yx} = 2 \text{Re} \left\{ (\hat{A} P A^H R^{-1} \times (\hat{A} P A^H R^{-1})^T
+ (\hat{A} P A^H R^{-1} \hat{A} P A^H R^{-1}) \times (R^{-1} \times R^{-1}) \right\} \tag{A.30}
\end{equation}
\begin{equation}
J_{yy} = 2 \text{Re} \left\{ (\hat{A} P A^H R^{-1} \times (\hat{A} P A^H R^{-1})^T
+ (\hat{A} P A^H R^{-1} \hat{A} P A^H R^{-1}) \times (R^{-1} \times R^{-1}) \right\}. \tag{A.31}
\end{equation}
Finally, using the fact that the FIM is a symmetric matrix, we have
\begin{equation}
J_{xy} = J_{yx}^T, \tag{A.32}
\end{equation}
\begin{equation}
J_{yy} = J_{yy}^T. \tag{A.33}
\end{equation}
These results can be used for computing the exact Cramer–Rao lower bound.

**References**


Anthony J. Weiss (S'84–M'86–SM'86) was born in London, England, in 1951. He received the B.Sc. (cum laude) degree from the Technion—Israel Institute of Technology, Haifa, in 1973 and the M.Sc. (summa cum laude) and Ph.D. (summa cum laude) degrees from Tel-Aviv University, Tel-Aviv, Israel, in 1982 and 1985, respectively, all in electrical engineering.

From 1973 to 1983 he was involved in research and development of numerous projects in the fields of communications, tracking systems, command and control, and emitter localization. In 1985 he became a Faculty member of the Department of Electronic Systems, Tel-Aviv University. He was also a consultant for the Department of Defence, for the EW Department of Tadiran Inc., and for the ELINT Department of Elta Electronics—a subsidiary of Israel Aircraft Industries (IAI). During the academic year 1986–1987 he was a Visiting Scientist with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge. In 1987–1988 he was with Saxby Computer Corporation, Sunnyvale, CA. His research activities are detection and estimation theory, signal processing, array processing, image processing, and parallel processing architectures, with applications to radar, sonar, and passive surveillance systems.


Benjamin Friedlander (S’74–M’76–SM’82–F’87), for a photograph and biography, see p. 179 of the February 1989 issue of this Transactions.