Title
Estimating two-dimensional frequencies by matrix enhancement and matrix pencil

Permalink
https://escholarship.org/uc/item/71d8k09w

Journal
IEEE Transactions on Signal Processing, 40(9)

ISSN
1053-587X

Author
Hua, Yingbo

Publication Date
1992-09-01

Peer reviewed
Estimating Two-Dimensional Frequencies by Matrix Enhancement and Matrix Pencil

Yingbo Hua, Senior Member, IEEE

Abstract—A new method, called the matrix enhancement and matrix pencil (MEMP) method, is presented for estimating two-dimensional (2-D) frequencies. The MEMP method first constructs an enhanced matrix from the data samples, and then uses the matrix pencil approach to extract out the 2-D sinusoids from the principal eigenvectors of the enhanced matrix. The MEMP method yields the estimates of the 2-D frequencies efficiently, without solving the roots of a 2-D polynomial or searching in a 2-D space. It is shown that the MEMP method can be faster than a 2-D FFT method if the number of the 2-D sinusoids is much smaller than the data set. Simulation results are provided to show that the accuracy of the MEMP method can be very close to the Cramér-Rao lower bound.

1. Introduction

In many applications, such as synthetic aperture radar imaging, frequency and wave-number estimation in array processing, and nuclear magnetic resonance imaging, it is often desired to estimate two-dimensional (2-D) frequencies from a 2-D data set. If the data set is very large, the classical correlogram method (implemented via 2-D FFT) can be satisfactory. If the data set is relatively small, the correlogram method suffers from a resolution limit called Rayleigh limit. To overcome the Rayleigh limit, high-resolution techniques such as the 2-D autoregressive method, 2-D maximum entropy method, and 2-D minimum variance method have been developed from their 1-D versions. To obtain the estimates of the 2-D frequencies, searching for spectral peaks in a 2-D space is required by all those methods. The searching is due to the difficulty in finding the desired roots of a 2-D polynomial. It causes a very large amount of computations, and hence limits the estimation accuracy given a fixed amount of computations. Tutorial discussions on those methods are available in [1] and [17].

The computational difficulty of searching in a 2-D space also exists with other methods in [9]–[11]. This is again due to the bottleneck: 2-D polynomial. A recently published method [12] for estimating 2-D frequency, although computationally efficient, only applies to the single 2-D sinusoid case.

A computationally efficient method for estimating multiple 2-D frequencies is available in the work by Kung et al. [2]. This method, called the state space method, does not require searching in a 2-D space. It exploits the structure inherent in the original data matrix. The 2-D frequencies are computed by solving an eigenvalue problem. However, the state space method does not work for the case where more than one 2-D sinusoids share a common 1-D sinusoidal component. Furthermore, this method yields two sets of estimated 1-D frequencies rather than a set of estimated 2-D frequencies. How to pair the two sets of estimated 1-D frequencies into a set of estimated 2-D frequencies was not mentioned in [2].

Another 2-D frequency estimation method, called the matrix approximation method, was proposed by Shaw-Kumarasen in [3]. Similar to the state space method in [2], the matrix approximation method is based on the inherent structure of the original data matrix. The difference is, however, that the matrix approximation method tries to reconstruct such a matrix, subject to the constraint of the known data structure, that approximates the original data matrix in a least square (LS) sense. Like those of [2], the authors of [3] did not address the problem arising from multiple 2-D frequencies having a common 1-D frequency component. The pairing issue was also not addressed in [3].

In this paper, we will follow an approach similar to those in [2] and [3] to exploit the structure inherent in the original data. However, instead of relying on the original data matrix, we will form an enhanced matrix from the original data. The enhanced matrix is formed in such a way that the matrix pencil approach in [4] can be applied to efficiently estimate the 2-D frequencies. The resulting method will be called the matrix enhancement and matrix pencil (MEMP) method.

In Section II, the 2-D frequency estimation problem will be formulated. A basic structure inherent in the original data matrix will be reviewed. It will be pointed out why the methods in [2] and [3] fail to work.

In Section III, the idea of matrix enhancement will be introduced, and the structure of an enhanced matrix will be studied. It will be shown that the number of 2-D sinusoids can be obtained from the rank of the enhanced matrix, and the 2-D frequencies can be obtained from the principal eigenvectors of the enhanced matrix.

In Section IV, the matrix pencil approach will be applied to efficiently estimate the 2-D frequencies from the principal eigenvectors of the enhanced matrix. The pairing issue will also be addressed.
In Section V, the noisy data will be assumed and the MEMP method is summarized into a step-by-step algorithm. For the case where the data set is very large and the noise covariance sequence is known except a scalar, an asymptotically consistent version will also be given.

\[
X = \begin{bmatrix}
  x(0; 0) & x(0; 1) \\
  x(1; 0) & x(1; 1) \\
  \vdots & \vdots \\
  x(M - 1; 0) & x(M - 1; 1) \\
\end{bmatrix}
\]  \hspace{1cm} \cdots x(0; N - 1) \\
\cdots x(1; N - 1) \\
\cdots \\
\cdots x(M - 1; N - 1)  \hspace{1cm} \]  \hspace{1cm} \]  \hspace{1cm} \]

Using (2.2) in (2.6) yields

\[
X = YAZ  \]  \hspace{1cm} \]

where

\[
Y = \begin{bmatrix}
  y_1 & y_2 & \cdots & y_I \\
  \vdots & \vdots & \ddots & \vdots \\
  y_{M^1} & y_{M^2} & \cdots & y_{M^I} \\
\end{bmatrix}
\]

\[
A = \text{diag} \{ a_1, a_2, \cdots, a_I \}
\]

\[
Z = \begin{bmatrix}
  z_1 & z_2 & \cdots & z_{I^1} \\
  1 & z_2 & \cdots & z_{I^2} \\
  \vdots & \vdots & \ddots & \vdots \\
  1 & z_I & \cdots & z_{I^I} \\
\end{bmatrix}
\]

From (2.7)-(2.10), we know that the rank of \(X\) is no larger than \(I\), i.e., \(\text{rank} (X) \leq I\). Due to the Vandermonde structure in \(Y\) and \(Z\), it can be shown that \(X\) has the rank \(I\) if and only if (iff) the two sets of the 1-D poles: \(\{ y_i; i = 1, \cdots, I \}\) and \(\{ z_i; i = 1, \cdots, I \}\) both contain distinct (nonzero) elements, provided \(M \geq I\) and \(N \geq I\). It was a basic condition under which the matrix space method [2] and the matrix approximation method [3] were developed. But the rank of \(X\) is less than \(I\) if either one of the pole sets does not contain distinct elements. Note that the assumption \((y_i, z_i) \neq (y_j, z_j)\) does not necessarily mean that \(y_i \neq y_j\) and \(z_i \neq z_j\). It is the ill condition (insufficient rank) of \(X\) that causes the two methods to fail.

It is important to note that a) if \(\text{rank} (X) \leq I\), \(\{ y_i; i = 1, \cdots, I \}\) and \(\{ z_i; i = 1, \cdots, I \}\) cannot be both obtained from the principal left or right singular vectors of \(X\); and b) the principal singular vectors of \(X\) do not contain sufficient information to carry out the pairing between \(y_i\) and \(z_i\).

In the next section, we will form an enhanced matrix from the 2-D data so that the above problems can be solved.

III. MATRIX ENHANCEMENT

The idea of the matrix enhancement can be seen from two simple examples as follows.

\[
X = \begin{bmatrix}
  x(0; 0) & x(0; 1) \\
  x(1; 0) & x(1; 1) \\
  \vdots & \vdots \\
  x(M - 1; 0) & x(M - 1; 1) \\
\end{bmatrix}
\]  \hspace{1cm} \cdots x(0; N - 1) \\
\cdots x(1; N - 1) \\
\cdots \\
\cdots x(M - 1; N - 1)  \hspace{1cm} \]  \hspace{1cm} \]  \hspace{1cm} \]

\[
Y = \begin{bmatrix}
  y_1 & y_2 & \cdots & y_I \\
  \vdots & \vdots & \ddots & \vdots \\
  y_{M^1} & y_{M^2} & \cdots & y_{M^I} \\
\end{bmatrix}
\]

\[
A = \text{diag} \{ a_1, a_2, \cdots, a_I \}
\]

\[
Z = \begin{bmatrix}
  z_1 & z_2 & \cdots & z_{I^1} \\
  1 & z_2 & \cdots & z_{I^2} \\
  \vdots & \vdots & \ddots & \vdots \\
  1 & z_I & \cdots & z_{I^I} \\
\end{bmatrix}
\]
Example 1: A row vector cannot have a rank larger than one. However, if this vector is partitioned into two (overlapped or nonoverlapped) subvectors and the two subvectors are stacked into a two-row matrix, then the resulting matrix may have a rank larger than one.

Example 2: If an m-row matrix has the rank r, the matrix obtained from a similar partition-and-stacking process may have a rank larger than r.

We see that the rank condition of a matrix can be enhanced by a partition-and-stacking process. (This idea is similar, from a mathematical point of view, to the ideas of moving window [19] for a uniform linear array problem and focusing [20] for a wide-band wave direction finding problem.)

Note that in this and the next sections, only the noiseless data samples will be considered. The noise effects will be discussed in Sections V and VII.

A. Enhanced Matrix $X_e$

An enhanced matrix useful for the 2-D frequency estimation problem is defined through a partition-and-stacking process as follows:

$$X_e = \begin{bmatrix} X_0 & X_1 & \cdots & X_{M-K} \\ X_1 & X_2 & \cdots & X_{M-K+1} \\ \vdots & \vdots & \ddots & \vdots \\ X_{K-1} & X_K & \cdots & X_{M-1} \end{bmatrix}$$  

(3.1)

where

$$X_m = \begin{bmatrix} x(m; 0) & x(m; 1) & \cdots & x(m; N - L) \\ x(m; 1) & x(m; 2) & \cdots & x(m; N - L + 1) \\ \vdots & \vdots & \ddots & \vdots \\ x(m; L - 1) & x(m; L) & \cdots & x(m; N - 1) \end{bmatrix}$$  

(3.2)

$X_e$ is an $K \times (M - K + 1)$ Hankel block matrix, and $X_m$ is an $L \times (N - L + 1)$ Hankel matrix. Each column of $X_m$ is a windowed segment of the sequence $\{x(m; 0), x(m; 1), \cdots, x(m; N - 1)\}$ with the window length $L$. Each column of $X_e$ is a windowed segment of the matrix sequence $\{X_0, X_1, \cdots, X_{M-1}\}$ with the window length $K$. In fact, $X_e$ is a generalized version of $X$. If $L = 1$ and $K = M$, $X_m$ becomes the $m$th row of $X$, and $X_e$ becomes $X$.

We call $X_e$ the enhanced matrix because

$$\text{rank}(X_e) = I \geq \text{rank}(X)$$  

(3.3)

in some important cases. The second relation in (3.3) has been discussed in Section II. The first relation in (3.3) will be discussed as follows.

Using (2.2) in (3.2), $X_m$ becomes

$$X_m = Z_L AZ_R$$  

(3.4)

where $Z_L$ is defined in (2.9),

$$Z_L = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_L \\ \vdots & \vdots & \ddots & \vdots \\ z_{L-1} & z_{L-2} & \cdots & z_1 \end{bmatrix}$$  

(3.5)

$$Y_d = \text{diag}(y_1, y_2, \cdots, y_L)$$  

(3.6)

$$Z_R = \begin{bmatrix} 1 & z_1 & \cdots & z_{N-L} \\ z_2 & z_1 & \cdots & z_{N-L} \\ \vdots & \vdots & \ddots & \vdots \\ z_L & z_{L-1} & \cdots & z_1 \end{bmatrix}$$  

(3.7)

Then, using (3.4) in (3.1), $X_e$ becomes

$$X_e = E_L A E_R$$  

(3.8)

where

$$E_L = \begin{bmatrix} Z_L \\ Z_1 Y_d \\ \vdots \\ Z_{K-1} Y_d^{K-1} \end{bmatrix}$$  

(3.9)

$$E_R = [Z_L, Z_1 Z_R, \cdots, Y_{M-K} Z_R]$$  

(3.10)

From (3.8), we know that rank $(X_e) = I$ iff rank $(E_L) = \text{rank}(E_R) = I$.

B. Conditions on K and L

Now we need to find the conditions on the free parameters $K$ and $L$ under which rank $(E_L) = \text{rank}(E_R) = I$.

Since the structures of $E_L$ and $E_R$ are similar, only $E_L$ is considered for the moment. Obviously, the rank of $E_L$ depends on the two parameters $K$ and $L$. We will show that rank $(E_L) = I$ if

$$K \geq 1 \quad \text{and} \quad L \geq 1$$  

(3.11)

To show this, we need to introduce the permutation (shuffling) matrix:

$$P = \begin{bmatrix} p(1) \\ p(1 + L) \\ \vdots \\ p(1 + (K - 1)L) \\ p(2) \\ p(2 + L) \\ \vdots \\ p(2 + (K - 1)L) \\ \vdots \\ p(L) \\ p(L + L) \\ \vdots \\ p(L + (K - 1)L) \end{bmatrix}$$  

(3.12)
where \( p(i) \) is the \( KL \times 1 \) vector with one at the \( i \)th position and zero everywhere else. The superscript \( \dagger \) denotes the transposition. Left multiplying \( E_i \) by \( P \) (i.e., shuffling the rows of \( E_i \)) yields \( E_{i,P} = PE_i \) which can be shown to be

\[
E_{i,P} = \begin{bmatrix}
Y_L \\
Y_I Z_d \\
\vdots \\
Y_I Z_d^{K-1}
\end{bmatrix}
\]  

(3.13)

where

\[
Y_L = \begin{bmatrix}
y_1 & y_2 & \cdots & y_I \\
1 & 1 & \cdots & 1 \\
y_I & y_2 & \cdots & y_I \\
\vdots & \vdots & \ddots & \vdots \\
y_I^{K-1} & y_2^{K-1} & \cdots & y_I^{K-1}
\end{bmatrix}
\]  

(3.14)

\[
Z_d = \text{diag}(z_1, z_2, \cdots, z_l).
\]

(3.15)

Note that the position of \( z_i \) in \( E_i \) is like that of \( y_i \) in \( E_{i,P} \), and the position of \( y_i \) in \( E_i \) is like that of \( z_i \) in \( E_{i,P} \). Since \( Z_d \) is a submatrix of \( E_i \), and \( Y_L \) is a submatrix of \( E_{i,P} \) (a shuffled version of \( E_i \)),

\[
\text{rank}(E_L) \geq \text{rank}\left[\begin{bmatrix}Z_d \\ Y_L\end{bmatrix}\right].
\]  

(3.16)

Since \( \{y_i, z_i; i = 1, 2, \cdots, l\} \) are distinct, the \( l \) columns of \( [Y_L] \) are linearly independent provided \( L \geq I \) and \( K \geq I \) (so that \( Z_d \) and \( Y_L \) each have no less than \( I \) rows). Hence, the sufficient condition (3.11) is proven.

The necessary condition for \( E_i \) to be of the full rank \( I \) is that the number of rows of \( E_i \) is no less than \( I \), i.e.,

\[
\text{rank}(E_i) = I \quad \text{only if}
\]

\[
KL \geq I. 
\]  

(3.17)

If \( K \) and \( L \) satisfy the necessary condition (3.17) but not the sufficient condition (3.11), \( \text{rank}(E_i) \) may or may not be equal to \( I \).

Due to the similarity between \( E_i \) and \( E_R \), it can be similarly shown that \( \text{rank}(E_R) = I \) if

\[
M - K + 1 \geq I \quad \text{and} \quad N - L + 1 \geq I
\]  

(3.18)

or only if

\[
(M - K + 1)(N - L + 1) \geq I. 
\]  

(3.19)

Since \( \text{rank}(X_e) = I \) if \( \text{rank}(E_i) = I \), combining (3.11) with (3.18) yields that \( \text{rank}(X_e) = I \) if

\[
M - I + 1 \geq K \geq I \quad \text{and} \quad N - I + 1 \geq L \geq I
\]  

(3.20)

and combining (3.17) with (3.19) yields that \( \text{rank}(X_e) = I \) only if

\[
KL \geq I \quad \text{and} \quad (M - K + 1)(N - L + 1) \geq I.
\]  

(3.21)

If \( I \) is unknown but less than a number \( I_{\text{max}} \), and \( K \) and \( L \) must satisfy the sufficient condition (3.20) so that \( I \) can be estimated from the singular values of \( X_e \). In the sequel (except in Section V), however, the number \( I \) will be assumed to be known.

C. Eigenstructure of \( X_e \)

Before we apply the matrix pencil approach, in the next section, to extract the 2-D poles from \( X_e \), we need to study the eigen structure of \( X_e \). The singular value decomposition (SVD) [5] of \( X_e \) has the form

\[
X_e = \sum_{i=1}^{\minKL} \sigma_i u_i v_i^H
\]

(3.22)

where the superscript \( H \) denotes the conjugate transpose; \( \min = \min(KL, (M - K + 1)(N - L + 1)) \) which is the smaller dimension of \( X_e \); \( U_i \), \( \Sigma_i \), and \( V_i \) contain the principal components; and \( U_{\text{min}}, \Sigma_{\text{min}}, \) and \( V_{\text{min}} \) contain the remaining nonprincipal components. Specifically,

\[
U_i = [u_1, u_2, \cdots, u_I]
\]  

(3.22a)

\[
\Sigma_i = \text{diag}([\sigma_1, \sigma_2, \cdots, \sigma_I])
\]  

(3.22b)

\[
V_i = [v_1, v_2, \cdots, v_I]
\]  

(3.22c)

\[
U_{\text{min}} = [u_{I+1}, u_{I+1}, \cdots, u_{\text{min}}]
\]  

(3.22d)

\[
\Sigma_{\text{min}} = \text{diag}([\sigma_{I+1}, \sigma_{I+2}, \cdots, \sigma_{\text{min}}])
\]  

(3.22e)

\[
V_{\text{min}} = [v_{I+1}, v_{I+2}, \cdots, v_{\text{min}}]
\]  

(3.22f)

where \( \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\text{min}} \). For the noiseless case, \( \sigma_i > 0 \) for \( i = 1, \cdots, I \), and \( \sigma_i = 0 \) for \( i > I \), and hence \( \Sigma \) is zero. Comparing (3.22) with (3.8) yields that if \( \text{rank}(X_e) = I \),

\[
\text{range}(X_e) = \text{range}(E_i) = \text{range}(U_i)
\]  

(3.23)

and

\[
\text{range}(X_e^H) = \text{range}(E_i^H) = \text{range}(V_i).
\]  

(3.24)

Since \( U_i \perp U_{\text{min}} \) and \( V_i \perp V_{\text{min}} \) where \( \perp \) denotes that the left is (columnwise) orthogonal to the right, \( E_i \perp U_i \) and \( E_i^H \perp V_i \). The above properties can be used to estimate the 2-D frequencies as follows. Note that only \( U_i \) will be used to produce the 2-D frequencies in the sequel although \( V_i \) can be used similarly. It seems that using both of the two matrices might yield better estimates, but such attempt has not been successful.

From the expression of \( E_i \) in (3.9), we know that the \( i \)th column of \( E_i \) is

\[
e_{yi} = y_{yi} \otimes z_{yi}
\]  

(3.25)

where \( \otimes \) denotes the Kronecker product, and \( y_{yi} \) and \( z_{yi} \) are the \( i \)th column of \( Y_L \) and \( Z_d \), respectively. Because of (3.25), we define a similar vector:

\[
e_i = y_i \otimes z_i
\]  

(3.26)

where

\[
y_i = [y_1, y_2, \cdots, y_I]^T
\]  

(3.27)

\[
z_i = [z_1, z_2, \cdots, z_L^{K-1}]^T
\]  

(3.28)
\[ y = \exp(j2\pi f_1) \]  
\[ z = \exp(j2\pi f_2). \]  
Clearly, \( e_L \) is a function of the 2-D frequency variable \((f_1, f_2)\). So, we may also write \( e_L(f_1, f_2) \) in place of \( e_L \) to emphasize the relationship. It can be shown that if \( K \) and \( L \) satisfy the sufficient condition (3.20), \( e_L \) belongs to span \( \{e_{l_1}, e_{l_2}, \ldots, e_{l_I}\} \) iff \((f_1, f_2) = (f_{l_1}, f_{l_2})\), and more importantly
\[ e_L \perp U \] (3.31)
iff \((f_1, f_2) = (f_{l_1}, f_{l_2})\). This suggests that \( \{f_{l_1}, f_{l_2}; i = 1, \ldots, I\} \) can be found from the peak positions of the 2-D frequency spectrum:
\[ \frac{1}{\min \sum_{i=1}^{I} ||U^H e_L(f_1, f_2)||^2} \] (3.32)
Estimating the 2-D frequencies by searching for the peaks of the 2-D spectrum of (3.32) is very costly in computation. The above approach is similar to the idea of MUSIC [11]. In the next section, we will use the matrix pencil approach to estimate the 2-D frequencies from the principal singular vectors of \( X \) (i.e., using \( U_3 \)).

IV. MATRIX PENCIL
The matrix pencil approach can be stated as constructing two matrices in such a way that the desired numbers (e.g., poles) are the rank reducing numbers (i.e., the generalized eigenvalues or G.E.s) of the corresponding matrix pencil.

A. Extracting \( y \)
We have shown that if the condition (3.20) is satisfied, range \( (U_1) = \text{range} (E_L) \), and hence
\[ U_1 = E_L T \] (4.1)
where \( T \) is an unique \( I \times I \) nonsingular matrix. (Note that both \( U_1 \) and \( E_L \) have \( I \) independent columns.) Knowing the structure of \( E_L \) shown in (3.9), we define
\[ U_1 = U_p \text{ with the last } I \text{ rows deleted} \] (4.2)
\[ U_2 = U_p \text{ with the first } I \text{ rows deleted} \] (4.3)
Using (4.1) and (3.9), we can write
\[ U_1 = E_1 T \] (4.4)
\[ U_2 = E_1 Y d T \] (4.5)
where
\[ E_1 = E_L \text{ with the last } I \text{ rows deleted}. \] (4.6)
Then, it is clear that the matrix pencil \( U_2 - \lambda U_1 \) becomes
\[ U_2 - \lambda U_1 = E_1 (Y d - \lambda I) T \] (4.7)
where \( I \) is an identity matrix of proper dimension. Since \( Y d \) is the diagonal matrix of the poles \( \{y_i; i = 1, \ldots, I\} \), (4.7) shows (see [4]) that the poles \( \{y_i; i = 1, \ldots, I\} \) are the rank reducing numbers of the matrix pencil \( U_2 - \lambda U_1 \) (i.e., the rank of the matrix pencil decreases by one iff \( \lambda = y_i \)), if \( E_1 \) and \( T \) are of the full rank \( I \).

B. Extracting \( z \)
In order to extract the other set of poles \( \{z_i; i = 1, \ldots, I\} \), we need to exploit the structure of \( E_{1p} \) in (3.13). We define
\[ U_{1p} = PU_i \] (4.8)
\[ U_{1p} = U_{1p} \text{ with the last } K \text{ rows deleted} \] (4.9)
\[ U_{2p} = U_{1p} \text{ with the first } K \text{ rows deleted.} \] (4.10)
Using (4.1) in (4.8), we have \( U_{1p} = PE_i T = E_{1p} T \). Using this result and (3.13) in both (4.9) and (4.10), we can write
\[ U_{1p} = E_{1p} T \] (4.11)
\[ U_{2p} = E_{1p} Z d T \] (4.12)
where
\[ E_{1p} = E_{1p} \text{ with the last } K \text{ rows deleted.} \] (4.13)
Then, we can write the matrix pencil \( U_{2p} - \lambda U_{1p} \) as
\[ U_{2p} - \lambda U_{1p} = E_{1p} (Z d - \lambda I) T \] (4.14)
Since \( Z d \) is the diagonal matrix of \( \{z_i; i = 1, \ldots, I\} \), (4.14) shows that \( \{z_i; i = 1, \ldots, I\} \) are the rank reducing numbers of \( U_{1p} - \lambda U_{1p} \), if \( E_{1p} \) and \( T \) are of the full rank \( I \).

C. Conditions on \( K \) and \( L \)
\( T \) has been known to be of the full rank \( I \) given \( K \) and \( L \), satisfying the condition (3.20). To find the conditions under which \( E_1 \) and \( E_{1p} \) are of the full rank \( I \), we should compare the two matrices with \( E_1 \) in (3.9). It can be shown (similarly as for (3.11)) that rank \( (E_1) = I \) if
\[ K - 1 \geq I \text{ and } L \geq I \] (4.15)
and rank \( (E_{1p}) = I \) if
\[ K \geq I \text{ and } L - 1 \geq I. \] (4.16)
Combining the above two conditions with (3.20), we obtain the overall sufficient condition (for the MEMP method to work):
\[
\begin{cases}
M - I + 1 \geq K \geq I + 1 \\
N - I + 1 \geq L - I + 1.
\end{cases}
\] (4.17)
On the other hand, the necessary condition for rank \( (E_1) \) to be \( I \) is that the number of its rows is no less than \( I \), i.e.,
\[ (K - 1)L \geq I. \] (4.18)
Similarly, the necessary condition for rank \( (E_{1p}) \) to be \( I \) is
\[ K(L - 1) \geq I. \] (4.19)
Combining the necessary conditions (4.18) and (4.19) and (3.21) yields the overall necessary condition:

\[
\begin{align*}
(K - 1)L & \geq I \\
K(L - 1) & \geq I \\
(M - K + 1)(N - L + 1) & \geq I.
\end{align*}
\]

(4.20)

D. Pairing

We have now developed the MEMP method to extract \{y_i; i = 1, \cdots, I\} and \{z_i; i = 1, \cdots, I\} separately. The order of poles in each set is still unknown. Note that the \(y_i\) in \{y_i; i = 1, \cdots, I\} is not necessarily the \(y_j\) in the (correct) pairs \{(y_i, z_j); i = 1, \cdots, I\}. To obtain the (correct but not necessarily ordered) pairs \{(y_i, z_j); i = 1, \cdots, I\}, we need to pair the two sets \{y_i; i = 1, \cdots, I\} and \{z_i; i = 1, \cdots, I\} together correctly. An optimum pairing approach may be such that a cost function of \{(y_i, z_j); i = 1, \cdots, I\} is minimized by a choice among \(I!\) (factorial) possibles. The cost function may be the sum of squared errors between the original data samples and the reconstructed data samples. However, if \(I\) is moderately large, \(I!\) can be too large to carry out the pairing process. To speed up the pairing process, we suggest to use the property shown in (3.31). Specifically, we do the following. For \(i = 1, 2, \cdots, I\), we minimize

\[
J_n(i, j) = \sum_{i=1}^{\min} \|u_i^{(j)} e_i(y_i, z_j)\|^2
\]

(4.21)

with respect to \(j\), where \(e_i(y_i, z_j) = e_i(f_{ji}, f_{jz})\). After expressing the algorithm in terms of the \(I\) principal eigenvectors \{\(u_i; i = 1, \cdots, I\}\}, i.e., using \(U_i U_i^H = I - U_i U_i^H\), we can equivalently maximize, for \(i = 1, 2, \cdots, I\),

\[
J_i(i, j) = \sum_{i=1}^I \|u_i^{(j)} e_i(y_i, z_j)\|^2
\]

(4.22)

with respect to \(j\) to find the proper pairing.

The above maximization can be explicitly carried out as follows.

1) Set \(i = 1\).
2) Compute \(J_i(i, j)\) for \(j = 1, 2, \cdots, I\).
3) Search for the largest value among \(\{J_i(i, j); j = 1, 2, \cdots, I\}\) to obtain the pairing index \(i,j(i)\); and hence the pair \(\{y_i, z_{j(i)}\}\).
4) Set \(i = i + 1\).
5) Compute \(J_i(i, j)\) for \(j = 1, 2, \cdots, I\) but \(j \neq j(k)\) where \(k = 1, 2, \cdots, i - 1\).
6) Search for the largest value among \(\{J_i(i, j); j = 1, 2, \cdots, I\}\) but \(j \neq j(k)\) where \(k = 1, 2, \cdots, I - 1\) to obtain the pairing index \(i,j(i)\) and hence the pair \(\{y_i, z_{j(i)}\}\).
7) Go to Step 4) unless \(i = I - 1\).

This procedure does not treat the poles \{\(y_i; i = 1, 2, \cdots, I\}\) equally, i.e., \(y_i\) is treated more seriously (earlier) than \(y_{i+1}\). If we have some \(a\ priori\) information about \(y_i\), then we may be able to order them according to their priority before the above pairing procedure is applied. If we know more about \(z_i\) than \(y_i\), then we should interchange the order of \(z_i\) and \(y_i\) in the above pairing procedure. The overall idea here is to ensure that the pole we have the best confidence with gets the best mate.

V. The MEMP Algorithms

The MEMP method for estimating 2-D frequencies has been developed in the previous two sections assuming no noise. In the following, we will first summarize the MEMP method into an algorithm which processes the noisy data. Then, another modified algorithm will be developed for the case where the data set is large and the noise covariance sequence is known.

A. Algorithm 1

Step 1: Form the \(KL \times (M - K + 1)(N - L + 1)\) enhanced matrix \(X_e\) from the noisy data \(x(m; n)\) according to (3.1). \(K\) and \(L\) must satisfy the necessary condition (4.20). For more reliable and more accurate estimation, \(K\) and \(L\) should satisfy the sufficient condition (4.17). (The effects of \(K\) and \(L\) on the noise sensitivity will be discussed in Section VII.)

Step 2: Compute the singular values and left singular vectors of \(X_e^H\). Estimate the number \(I\) of 2-D sinusoids from the singular values. (e.g., see [6]). Let \(U_i\) be the matrix of the \(I\) principal left singular vectors (as \(U_i\) in 3.22a). \(U_i\).

Step 3: Form \(U_{ip}\) from \(U_i\) according to (4.2) and (4.3). Form \(U_{ip}\) from \(U_{ip}\) according to (4.8). Form \(U_{ip}\) from \(U_{ip}\) according to (4.9) and (4.10).

Step 4: Compute the generalized eigenvalues (GE's) of \(U_i - \lambda U_i\) and \(U_{ip} - \lambda U_{ip}\). Let \{\(y_i; i = 1, \cdots, I\}\) be the first set of GE's, and \{\(z_i; i = 1, \cdots, I\}\) the second. (Several algorithms for the generalized eigenvalue problem are available in [7] or [13]–[15]. The simplest is to use the QZ algorithm [5] to solve the GE's of \(U_i^H U_i - \mu U_i^H U_i\) and \(U_{ip}^H U_{ip} - \mu U_{ip}^H U_{ip}\). The simulation results shown in Section VII were obtained by using this algorithm.)

Step 5: For \(i = 1, 2, \cdots, I\), maximize the function shown in (4.22) (using \(y_i^c\) and \(z_i^c\) in place of \(y_i\) and \(z_i\) respectively) with respect to \(j\) to obtain the correct pairs \{(\(y_i, z_j\); \(i = 1, \cdots, I\)}

Step 6: Compute \((f_{ij}, f_{jz})^T\) from \((y_i, z_j)\) by using (2.3) and (2.4).

B. Algorithm 2

If the data set is very large and the noise covariance sequence is known, then the noise can be filtered out at the covariance level as follows.

Let \(R_e^c\) be the covariance matrix of the noisy enhanced matrix \(X_e\), i.e.,

\[
R_e^c = \frac{1}{c} X_e X_e^H
\]

(5.1)
where \( c = (M - K + 1)(N - L + 1) \). If \((M - K)\) and \((N - L)\) are very large, then
\[
R_c' \equiv R_c + \gamma R_{cn}
\]
where \(R_c\) is the covariance matrix of the noiseless matrix \(X_c\), i.e.,
\[
R_c = \frac{1}{c} X_c X_c^H
\]
and \(\gamma R_{cn}\) is the noise covariance matrix, and \(\gamma\) is a scalar. Note that \(\gamma R_{cn}\) is a known function, although not given here, of the 2-D covariance sequence of the 2-D noise sequence \(w(m; n)\).

Assume \(R_{cn}\) is known. Then the effect of \(\gamma R_{cn}\) on the estimation accuracy can be removed (completely if \(M - K\) and \(M - L\) are infinitely large and the noise is stationary and ergodic) by using the following algorithm.

**Step 1:** Same as in algorithm 1 but then compute \(R_c'\) and its transformed version:
\[
R_c'^{\prime} = R_c^{\prime \prime \prime} R_c R_c^{-H/2^2}.
\]

**Step 2:** Compute the eigendecomposition of \(R_c'\), i.e.,
\[
R_c' = \sum_{i=1}^{KL} \lambda_i' u_i' u_i''^H.
\]

Estimate \(I\) from the eigenvalues \(\lambda_1' \geq \lambda_2' \geq \cdots \geq \lambda_{KL}'\). (The information criteria shown in [8] can be applied.) Note that asymptotically, \(\lambda_1' \geq \gamma\) for \(i = 1, \cdots, I\), and \(\lambda_{KL} = \gamma\) for \(i = I + 1, I + 2, \cdots, KL\). Let \(U_i'\) be the matrix of the \(I\) principal eigenvectors of \(R_c'\). Then, compute
\[
U_i' = R_c^{1/2} U_i' u_i''.
\]

**Steps 3–6:** Same as in algorithm 1 but use \(U_c''\) in place of \(U_c'\).

**C. Remarks**

If the noise is white, then it is easy to show that \(R_{cn} = I\) and hence algorithm 2 is equivalent to algorithm 1.

In some applications, not both \(M\) and \(N\) are very large, but one of them is. In this case, algorithm 2 is still applicable.

Finally, we mention that the MEMP method also applies to damped 2-D sinusoids since the data structure we have exploited so far is shown in (2.2). Given that all the poles \(y_i\) and \(z_i\) are on the unit circle for the undamped 2-D sinusoids, the enhanced matrix \(X_c\) can be replaced by a further enhanced matrix \(X_{ce}\):
\[
X_{ce} = \{X_c, P_c X_c^H\}
\]
where \(*\) denotes the complex conjugation, and \(P_c\) is a permutation matrix defined by
\[
P_c = \begin{bmatrix}
1 & & \\
\vdots & 1 & \\
& & 1
\end{bmatrix}.
\]

In the noiseless case,
\[
\text{range } (X_c) = \text{range } (X_c^H) = \text{range } (E^H)
\]
so that the MEMP method based on either \(X_c\) or \(X_c^H\) yields the same results. But in the noisy case, using \(X_c\) enhances the robustness to noise in a similar way as using the forward-and-backward linear prediction equations in [16] or as using the forward-and-backward matrix pencil in [4]. The simulation results shown in Section VII were obtained by using \(X_c\).

**VI. COMPUTATIONAL ORDER OF THE MEMP METHOD**

In the following, we will first derive an estimate of the order of real multiplications needed by each major computational part of the MEMP algorithm 1 for 2-D frequency estimation. The real data is assumed although the notations for complex data are used. Then, we will compare the computational order of the MEMP method against that of a 2-D FFT method.

The major computations required by the MEMP algorithm 1 are a) computing the singular values and the left singular vectors of \(X_c\), which has the dimension \(KL \times (M - K + 1)(N - L + 1)\); b) computing the generalized eigenvalues (GE's) of the matrix pencils \(U_i - \lambda U_i\) and \(U_{ip} - \lambda U_{ip}\) both of which have the dimension \(KL \times I\); and c) computing all necessary \(J_i(i, j)\) to pair \(\{y_i\}\) and \(\{z_i\}\).

**A. Computational Order of Major Part a)**

According to the Chan SVD [5], computing the singular values and left singular vectors of \(X_c\) requires
\[
K^2L^2(M - K + 1)(N - L + 1) \approx \frac{1}{2} K^2 L^3
\]
multiplications, where \((M - K + 1)(N - L + 1) > KL\).

However, the computational order shown in (6.1) can be reduced in the following approach is used to perform major part a):

1. Compute \(R_c = X_c X_c^H\) (in a fast way); and
2. Compute the eigenvalues and eigenvectors of \(R_c\).

Note that the left singular vectors of \(X_c\) are the eigenvectors of \(X_c X_c^H\), and the singular values of \(X_c\) are the square roots of the eigenvalues of \(X_c X_c^H\). (The numerical accuracy affected by the above approach is often negligible compared to the noise effects.)

A direct computation of \(X_c X_c^H\) requires
\[
\frac{1}{2} KL (KL + 1)(M - K + 1)(N - L + 1)
\]
multiplications, which is in order similar to that shown in (6.1).

To compute \(X_c X_c^H\) faster, we need to observe that the \((i_m, j_n)\)th element of the \((i_m, j_n)\)th block of \(X_c X_c^H\) (see (3.1)) is
\[
\sum_{k_m=-1}^{M-K-1} \sum_{k_n=-1}^{N-L-1} x(i_m + k_m; i_n + k_n) x^*(j_m + k_m; j_n + k_n)
\]
(6.3)
where \( 1 \leq i_m \leq K, 1 \leq j_m \leq K, 1 \leq i_n \leq l \) and \( 1 \leq j_n \leq l \). We can see that (almost) each multiplication in (6.3) is shared in computing several elements of \( X_n X^n_p \). In fact, this is due to the Hankel structure in \( X_n \). The distinct multiplications required in computing \( X_n X^n_p \) are

\[
x(m; n) = x(m - i_m, n - j_n)
\]

where \( m, n, t_m \), and \( t_n \) take the following sets of integers:

\[
\begin{align*}
\{t_m = 0; \{m = 0, 1, \cdots, M - 1; \{t_n = 0, 1, \cdots, L - 1; \{n = t_n, t_n + 1, \cdots, N - 1\}\}\}\}
\end{align*}
\]

and

\[
\begin{align*}
\{t_m = 1, 2, \cdots, K - 1; \{m = t_m, t_m + 1, \cdots, M - 1; \{t_n = 0, 1, \cdots, L - 1; \{n = t_n, t_n + 1, \cdots, N - 1\}\}\}\}
\end{align*}
\]

We can see that (almost) each multiplication in (6.3) is shared in computing several elements of \( X_n X^n_p \). In fact, this is due to the Hankel structure in \( X_n \). The distinct multiplications required in computing \( X_n X^n_p \) are

\[
x(m; n) = x(m - i_m, n - j_n)
\]

where \( m, n, t_m \), and \( t_n \) take the following sets of integers:

\[
\begin{align*}
\{t_m = 0; \{m = 0, 1, \cdots, M - 1; \{t_n = 0, 1, \cdots, L - 1; \{n = t_n, t_n + 1, \cdots, N - 1\}\}\}\}
\end{align*}
\]

Note that, for example, \( \{t_m = 1, 2, \cdots, K - 1; \{m = t_m, t_m + 1, \cdots, M - 1\} \} \) means that for each value of \( t_m \) taken from \( \{1, 2, \cdots, K - 1\} \), \( m \) takes all values from \( \{t_m, t_m + 1, \cdots, M - 1\} \). The Hermitian property of \( X_n X^n_p \) and its diagonal blocks has been considered to obtain the above result.

Now, the minimum number of multiplications required to compute \( X_n X^n_p \) can be calculated as follows:

\[
\sum_{n=0}^{M-1} \sum_{i=0}^{N-1} + \sum_{t_m=1}^{K-1} \sum_{m=0}^{M-1} \sum_{i=0}^{L-1} \sum_{n=0}^{N-1} 1 = \frac{K-1}{2} M \left( M - K \right) \left( N - \frac{L}{2} \right)
\]

which can be simplified as follows, assuming that \( K \gg 1, L \gg 1, M - K \gg 1 \) and \( N - L \gg 1 \),

\[
2KL \left( \frac{M - K}{2} \right) \left( \frac{N - L}{2} \right).
\]

We can see that (6.6) is much smaller than (6.2).

To compute the eigenvalues and eigenvectors of \( \mathbf{R}_r \), the number of multiplications required (based on the symmetric QR algorithm [5] and \( KL \gg 1 \)) is in the order of

\[
5K^2 L^3.
\]

Combining (6.6) and (6.7) yields that a1) and a2) can be carried out by using

\[
2KL \left( \frac{M - K}{2} \right) \left( \frac{N - L}{2} \right) + 5K^2 L^3.
\]

D. Computational Order of the MEMP Method

Combining (6.8)--(6.10) yields the order of (real) multiplications required by the MEMP algorithm 1:

\[
2KL \left( M - K \right) \left( N - \frac{L}{2} \right) + 5K^2 L^3 + 3K^2 L^3 + \frac{3}{2} K^2 L^3
\]

provided \( K \gg 1, L \gg 1, M \gg 1 \). (Note that as an engineering approximation, the "much larger" notation \( \gg \) means at least ten times larger. We see that the computational order for the major part a) is dominant in the overall computational order of the MEMP method. We also see that if \( M \gg K \) and \( N \gg L \), the dominant computation is to obtain the covariance matrix \( \mathbf{R}_r \) (or \( \mathbf{R}_r^* \) in the noisy case).
E. Comparison to 2-D FFT Method

We consider a 2-D FFT method which typically requires
\[ \frac{1}{2} (\log_2 MN) MN \] (6.12)
multiplications to produce a 2-D frequency spectrum. Note that (6.12) does not include the amount of computations required to search for 1-D (peak) frequencies. It is clear that if \( I \) is much smaller than \( M \) and \( N \), then \( K \) and \( L \) of the MEMP method (see (4.17)) can also be much smaller than \( M \) and \( N \), and hence (6.11) can be smaller than (6.12). In fact, the ratio of the computational order of the MEMP method over the 2-D FFT method is, assuming that \( 1 \ll I \ll K \ll M \) and \( 1 \ll I \ll L \ll N \),
\[ \frac{4KL}{\log_2 MN}. \] (6.13)
This means that if the number of 2-D sinusoids is much smaller than the data size, the MEMP method can be faster than the 2-D FFT method.

VII. SIMULATION RESULTS

The MEMP method has been developed based on the noiseless data. Without noise, the MEMP method yields the exact 2-D frequencies. When the data is noisy, the estimated 2-D frequencies have bias and variance. To evaluate the noise sensitivity of the MEMP method, the Monte Carlo simulation has been carried out. We have found that the estimation accuracy of the MEMP method can be very close to the CRB when \( K \) and \( L \) are not close to their boundary values shown in (4.17). A set of equations useful for computing the CRB will be given in Appendix A. A typical example of our simulation results is as follows.

A set of 20 \( \times \) 20 data samples are
\[ x'(m; n) = \sum_{i=1}^{3} \exp (j2\pi f_i m + j2\pi f_i n) + w(m; n) \] (7.1)
where \( 0 \leq m \leq 19 \), and \( 0 \leq n \leq 19 \), and \( w(m; n) \) is the white noise sequence, and
\[ (f_{i1}, f_{i2}) = (0.26, 0.24) \] (7.2)
\[ (f_{i1}, f_{i2}) = (0.24, 0.24) \] (7.3)
\[ (f_{i1}, f_{i2}) = (0.24, 0.26). \] (7.4)
This data set consists of three 2-D sinusoids which are too close to each other to be resolved by 2-D FFT since the Rayleigh limit for a 20 \( \times \) 20 data set is 0.05. The three 2-D sinusoids can neither be resolved by the state space method [2] or the matrix approximation method [3] since the first two 2-D frequencies have the second 1-D frequency component in common and the last two 2-D frequencies have the first 1-D frequency component in common.

Although other methods as in [1], [9]-[11] can be applied to this data set to estimate the three 2-D frequencies, they are too costly in computation. The simulation comparison of the MEMP method against those methods has not been obtained.

Figs. 1-4 show the estimated frequencies for 200 independent runs at SNR = 20 dB, which were obtained by using the MEMP Algorithm 1 and \( X_n \). SNR is defined by
\[ \text{SNR} = 10 \log_{10} \frac{1}{\gamma} \] (7.5)
where \( \gamma \) is the variance of the complex white noise. For Figs. 1-4, the two parameters \( (K, L) \) are equal to (3, 3), (4, 4), (5, 5), and (6, 6), respectively. Due to the symmetry of the 2-D data samples, we have chosen \( K = L \). It is clear from Figs. 1-4 that as \( K \) and \( L \) increase, the
frequencies start to cluster around each of three correct 2-D frequencies. When \( K = L = 6 \), Fig. 8 shows three denser clusters around each correct position. When \( K = L = 7 \), Fig. 9 shows that the three clusters are further compressed towards each correct position. The biases and deviations for \( K = L = 7 \) are shown with the corresponding CRB's in Table II.

From Tables I and II we can see that the estimation deviations are very close to the corresponding CRB's. We also can see from Figs. 1–9 that \( K \) and \( L \) are like two tuning parameters which can be adjusted to increase the estimation accuracy. In fact, when \( KL > I \), there is a \((\min - I)\)-dimensional noise subspace spanned by the columns of \( U_r \) and an \( I\)-dimensional signal subspace spanned by the columns of \( U_i \) (see (3.22)). Intuitively, the larger the noise subspace is, the more noise component is absorbed into the noise subspace and the less noise component remains in the signal subspace. Since only the signal subspace is used in the MEMP method, larger noise subspace implies higher estimation accuracy. We can increase the noise subspace, and hence (intuitively) the estimation accuracy, by increasing \( \min \) or equivalently \( K \) and \( L \) if \( KL < (M - K + 1)(N - L + 1) \). This intuitive thinking explains the simulation results shown in Figs. 1–9. In fact, the above intuitive thinking represents a signal processing approach, which we call signal processing via inflating noise subspace (SPINS).

Note that the noise subspace is maximized when \( KL = (M - K + 1)(N - L + 1) \) or when \( K = \frac{1}{2}(M + 1) \) and \( L = \frac{1}{2}(N + 1) \). Also note that if \( KL > (M - K + 1)(N - L + 1) \), not only the estimation accuracy is reduced (due to that the noise subspace is reduced) but also the computations are increased. Hence, we should normally choose \( K \) and \( L \) such that

\[
\begin{align*}
\frac{1}{2}(M + 1) &\geq K \geq I + 1 \\
\frac{1}{2}(N + 1) &\geq L \geq I + 1.
\end{align*}
\]

As long as the computational burden is tolerable, \( K \) and \( L \) can be increased, from \( I + 1 \) to the maximum values \( \frac{1}{2}(M + 1) \) and \( \frac{1}{2}(N + 1) \), to reduce the noise effects.

VIII. CONCLUSIONS

We have developed the MEMP method for estimating 2-D frequencies. An enhanced matrix has been introduced to remove the ill condition of the original data matrix. The matrix pencil approach has been applied to efficiently extract out the 2-D frequencies. The MEMP method is computationally efficient mainly because searching in a 2-D space is not required. An estimated order of the computations required by the MEMP method has been derived. We have shown that if the number of 2-D sinusoids is much smaller than the size of the data set, the MEMP method can be faster than a 2-D FFT method.

The noise sensitivity of the MEMP method has been studied by the Monte Carlo simulation. The simulation...
results show that the MEMP method is robust to noise and its accuracy can be very close to the CRB.

Finally, we add that the MEMP method can be extended to estimate arbitrary dimensional frequencies. Further research is under way in this direction. A severe problem in 3-D frequency estimation associated with applications such as 3-D radar imaging is data missing in the collected 3-D data set.
Fig. 9. Two hundred independent estimates of three 2-D frequencies. \( K = L = 7 \). SNR = 10 dB.

**TABLE II**

BIASES AND DEVIATIONS OF 200 INDEPENDENT ESTIMATES OF THREE 2-D FREQUENCIES. THE CRB'S SHOWN HERE ARE THE CRB'S ON DEVIATIONS. \( K = L = 7 \). SNR = 10 dB

<table>
<thead>
<tr>
<th>( f_1 )</th>
<th>Bias ( \times 10^{-3} )</th>
<th>Dev ( \times 10^{-3} )</th>
<th>CRB ( \times 10^{-1} )</th>
<th>( f_2 )</th>
<th>Bias ( \times 10^{-3} )</th>
<th>Dev ( \times 10^{-3} )</th>
<th>CRB ( \times 10^{-1} )</th>
<th>( f_3 )</th>
<th>Bias ( \times 10^{-3} )</th>
<th>Dev ( \times 10^{-3} )</th>
<th>CRB ( \times 10^{-1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.26</td>
<td>0.22</td>
<td>0.33</td>
<td>0.13</td>
<td>0.24</td>
<td>0.24</td>
<td>0.06</td>
<td>0.11</td>
<td>0.10</td>
<td>0.26</td>
<td>0.22</td>
<td>0.18</td>
</tr>
<tr>
<td>0.24</td>
<td>-0.22</td>
<td>0.18</td>
<td>0.10</td>
<td>0.24</td>
<td>-0.29</td>
<td>0.16</td>
<td>0.10</td>
<td>0.10</td>
<td>-0.09</td>
<td>0.13</td>
<td>0.10</td>
</tr>
</tbody>
</table>

**APPENDIX A**

CRB FOR 2-D FREQUENCY ESTIMATION

For the 2-D frequency estimation problem, the CRB is not readily available in literature. A set of equations which can be easily used to compute the CRB are given in the following.

We write

\[
x = \text{vec} \{x(m; n); m, n\} \quad (A.1)
\]

\[
w = \text{vec} \{w(m; n); m, n\} \quad (A.2)
\]

where vec \{ \} denotes a vector filled with the corresponding elements. Since \( x'(m, n) = x(m, n) + w(m, n) \), we write

\[
x' = x + w. \quad (A.3)
\]

Assume that the noise \( w \) is the (complex) white Gaussian, then the probability density function of \( x' \) is

\[
p(x'|\theta) = \frac{1}{(2\pi\gamma)^{MN}} \exp \left( -\frac{1}{\gamma} \| x' - x \|^2 \right) \quad (A.4)
\]

where \( \| \| \) denotes the 2-norm, \( \gamma \) is the noise variance, and \( \theta \) is the \( 4I \times 1 \) vector of unknown parameters, i.e.,

\[
\theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_I \end{bmatrix} \quad (A.5)
\]

and

\[
\theta_j = \begin{bmatrix} f_{ij} \\ \phi_i \end{bmatrix} \quad (A.6)
\]

The corresponding \( 4I \times 4I \) Fisher information matrix \( F \) is defined as follows [21]:

\[
F_{ij} = -\mathbb{E} \left\{ \frac{\partial^2}{\partial \theta_i \partial \theta_j} \log(p(x'|\theta)) \right\} \quad (A.7)
\]

where \( F_{ij} \) is the \((i, j)\)th element of \( F \), \( \mathbb{E} \{ \} \) the expectation, \( (\partial / \partial \theta_j) \) the partial derivative with respect to the \( i \)th element \( \theta_i \) of \( \theta \), and \( \log(\cdot) \) the natural logarithm. It can be shown, using (A.4) in (A.7), that

\[
F_{ij} = \frac{1}{\gamma} 2 \text{ Re} \left[ \frac{\partial x^H}{\partial \theta_i} \frac{\partial x}{\partial \theta_j} \right] \quad (A.8)
\]

where Re \{ \} denotes the real part.

Using (2.1) in (A.1), we can show the following:

\[
\frac{\partial x}{\partial \theta_i} = \text{vec} \{ \exp[j(\phi_i + 2\pi f_{im} + 2\pi f_{jn})]; m, n \} \quad (A.9)
\]
\[ \frac{\partial x}{\partial \phi_i} = \text{vec} \{ j r_i \exp [j (\phi_i + 2 \pi f_i m + 2 \pi f_i n)] ; m, n \} \]
\[ (A.10) \]
\[ \frac{\partial x}{\partial f_{ij}} = \text{vec} \{ j 2 \pi n r_i \exp [j (\phi_i + 2 \pi f_i m + 2 \pi f_i n)] ; m, n \} \]
\[ (A.11) \]
\[ \frac{\partial x}{\partial f_{ji}} = \text{vec} \{ j 2 \pi m r_i \exp [j (\phi_i + 2 \pi f_i m + 2 \pi f_i n)] ; m, n \} \]
\[ (A.12) \]

By using (A.9)-(A.12), the following can be shown:

\[ 2 \Re \left[ \frac{\partial x^H}{\partial r_j} \frac{\partial x}{\partial r_j} \right] = 2 \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \cos (\phi_i - \phi_j) + 2 \pi (f_{ij} - f_{ij}) m + 2 \pi (f_{ij} - f_{ij}) n \]
\[ (A.13) \]

\[ 2 \Re \left[ \frac{\partial x^H}{\partial \phi_i} \frac{\partial x}{\partial \phi_i} \right] = 2 r_j \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \sin (\phi_i - \phi_j) + 2 \pi (f_{ij} - f_{ij}) m + 2 \pi (f_{ij} - f_{ij}) n \]
\[ (A.14) \]

\[ 2 \Re \left[ \frac{\partial x^H}{\partial r_j} \frac{\partial x}{\partial f_{ij}} \right] = 2 r_j 2 \pi \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} m \sin (\phi_i - \phi_j) + 2 \pi (f_{ij} - f_{ij}) m + 2 \pi (f_{ij} - f_{ij}) n \]
\[ (A.15) \]

\[ 2 \Re \left[ \frac{\partial x^H}{\partial \phi_i} \frac{\partial x}{\partial f_{ij}} \right] = 2 r_j \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \cos (\phi_i - \phi_j) + 2 \pi (f_{ij} - f_{ij}) m + 2 \pi (f_{ij} - f_{ij}) n \]
\[ (A.16) \]

\[ 2 \Re \left[ \frac{\partial x^H}{\partial r_j} \frac{\partial x}{\partial r_j} \right] = 2 r_j 2 \pi \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} n \cos (\phi_i - \phi_j) + 2 \pi (f_{ij} - f_{ij}) m + 2 \pi (f_{ij} - f_{ij}) n \]
\[ (A.17) \]

\[ 2 \Re \left[ \frac{\partial x^H}{\partial \phi_i} \frac{\partial x}{\partial f_{ij}} \right] = 2 r_j 2 \pi \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} m \cos (\phi_i - \phi_j) + 2 \pi (f_{ij} - f_{ij}) m + 2 \pi (f_{ij} - f_{ij}) n \]
\[ (A.18) \]

By using (A.13)-(A.22), in (A.8), the 4I \times 4I Fisher information matrix can be straightforwardly formed. The CRB on the variance of the unbiased estimate of the ith parameter \( \theta_i \) is the ith diagonal element of the inverse \( F^{-1} \) which can be computed numerically. The CRB's on the corresponding deviations are the square roots of the diagonal elements of \( F^{-1} \). The CRB's listed in Tables I, II are the CRB's on deviations.

**APPENDIX B**

**ESTIMATING AMPLITUDES**

The signal amplitudes \( \{ a_i; i = 1, \ldots, I \} \) can be simply obtained (among other ways) as follows. Once \( \{(y_i, z_i); i = 1, \ldots, I\} \) are known or estimated, \( E_t \) and \( E_k \) can be reconstructed according to (3.5) and (3.7), respectively. From (3.8), we know that

\[ A = E_t^* X e E_k \]
\[ (B.1) \]

where the superscript \( ^* \) denotes the Moore-Penrose inverse, and

\[ E_t^* = (E_t^H E_t)^{-1} E_t^H \]
\[ (B.2) \]

\[ E_k^* = E_k^H (E_k E_k^H)^{-1} \].
\[ (B.3) \]

\( \{ a_i; i = 1, \ldots, I \} \) are the diagonal elements of \( A \).

If the noise level is high, and the noise covariance matrix is known (except a scalar), and the data set is large, then we can do the following. It can be shown, using (5.2)
and (5.5), that the signal only covariance matrix $R_c$ can be obtained from $X_c$ as follows:

$$R_c = \sum_{i=1}^{L} (\lambda_i - \gamma) u_i u_i^H$$  \hspace{1cm} (B.4)

where the notations are defined in (5.5). Equation (B.4) is the asymptotical result so that the primes are dropped. In the asymptotical case, $\gamma = \lambda_0$ for $i = L + 1, \cdots, K_L$. Then, using (3.8) in (5.3), we obtain

$$\frac{1}{c} E_c A E_R E_H^H A^H R_c = R_c$$ \hspace{1cm} (B.5)

and hence

$$\frac{1}{c} A E_R E_H^H A^H = E_L^T R_c (E_L^H)^+$$ \hspace{1cm} (B.6)

Using (3.10), we can write

$$\frac{1}{c} E_R E_H^H = \frac{1}{c} \sum_{m=0}^{M-K} Y_m^H Z_m^H Y_m^H$$ \hspace{1cm} (B.7)

Since $N - L$ is assumed to be very large, using (3.7) yields

$$Z_m Z_m^H = (N - L + 1) I$$ \hspace{1cm} (B.8)

where $z_i$'s in $Z_R$ are assumed to be distinct. Using (B.8) in (B.7) leads to

$$\frac{1}{c} E_R E_H^H = I.$$ \hspace{1cm} (B.9)

Hence, combining (B.9) and (B.6) yields

$$A A^H = E_L^T R_c (E_L^H)^+$$ \hspace{1cm} (B.10)

The absolute amplitudes $|a_i|$ are obtained from the diagonal elements of (B.10), but the phases are lost in (B.10).

ACKNOWLEDGMENT

The author would like to acknowledge Dr. T. K. Sarkar's encouragement for this work.

REFERENCES


Yingbo Hua (S'86-M'88-SM'92) was born in Wu-Xi, Jiangsu, China, on November 26, 1960. He received the B.S. degree in control engineering from Southeast University (Nanjing Institute of Technology), Nanjing, Jiangsu, China, in 1982, and the M.S. and Ph.D. degrees in electrical engineering from Syracuse University, Syracuse, NY, in 1983 and 1988, respectively.

He was a Graduate Teaching Assistant from 1984 to 1985, a University Graduate Fellow from 1985 to 1986, a Graduate Research Assistant from 1986 to 1988, and a Postdoctoral Research Associate from 1988 to 1989, all at Syracuse University. Since 1990, he has been a Lecturer with the University of Melbourne, Victoria, Australia. He has contributed over 40 publications dealing with spectral analysis and array processing.