Asymptotic Performance Analysis of ESPRIT, Higher Order ESPRIT, and Virtual ESPRIT Algorithms

Norman Yuen and Benjamin Friedlander, Fellow, IEEE

Abstract—In this paper, we present an asymptotic performance analysis of three subspace-based methods for direction of arrival (DOA) estimation—the ESPRIT algorithm using second-order statistics, the higher order ESPRIT algorithm using fourth-order cumulants, and the virtual ESPRIT (VESPA) algorithm using fourth-order cumulants. We examine the least-squares version of these algorithms, derive the expressions for the asymptotic variance of the estimated DOA’s, and use specific examples to compare the relative performance of the algorithms. Finally, we present Monte Carlo simulations to validate the theoretical analysis.

I. INTRODUCTION

IN THE last 15 years, subspace-based methods for estimating the directions of arrival (DOA’s) of far-field sources impinging on an array of sensors have become very popular. In particular, the MUSIC algorithm and its derivatives have received much attention. Extensive simulations and performance analyses have been presented by several authors showing the benefits of MUSIC in array processing estimation problems. One disadvantage of these subspace-based methods, including MUSIC, is that information (e.g., gain, phase, relative location) about the array of sensors must be known a priori. The ESPRIT algorithm, developed by Roy et al. [9], [12], lessens the a priori information requirement by using two identical arrays of sensors.

The performance of second-order ESPRIT and its derivative algorithms has been analyzed by several authors. Ottersten et al. [8] derived the asymptotic distribution of the error in estimated DOA’s for the total least-squares ESPRIT (TLS-ESPRIT) and compared the variance to the Cramer–Rao bound. Mathews and Zolotowski [7] derived the asymptotic variance for the uniform circular array ESPRIT, which is a 2-D angle estimation algorithm. Gavish and Weiss [5] analyzed the virtual interpolated array ESPRIT algorithm, which is based on interpolated arrays [4]. Li et al. [6] derived expressions for the variance of estimated DOA’s for the ESPRIT algorithm using the data matrix.

The performance of ESPRIT has also been analyzed by Rao and Hari [11], who derived the mean squared error (MSE) for estimated DOA’s of a second-order ESPRIT algorithm. They assumed a linear equispaced array and jointly complex Gaussian sources in additive white Gaussian noise. Their analysis is done for the least-squares version of ESPRIT. However, they show that the asymptotic MSE’s of the LS-ESPRIT and TLS-ESPRIT are the same. Our work in this paper follows that of Rao and Hari.

While the analysis in [11] is valid for only the second-order ESPRIT and for only Gaussian sources, the development in this paper is valid for Gaussian and non-Gaussian sources in the second-order case. Furthermore, the analysis here encompasses the fourth-order ESPRIT [2] and the recently developed fourth-order VESPA [3] algorithms.

This paper is organized as follows. Section II summarizes the least-squares version of the second-order ESPRIT, fourth-order ESPRIT, and VESPA algorithms. The asymptotic variance of the DOA estimates is then derived in Section III. The analysis common to all three ESPRIT algorithms is first presented and then the analysis for each specific algorithm is given. In Section IV, Monte Carlo simulations and numerical evaluations of the equations derived in the previous section are provided to show the match between analysis and simulation. This paper concludes in Section V. Some relevant equations are derived in the Appendix.

II. SUMMARY OF ESPRIT ALGORITHMS

In this section, we summarize the least-squares (LS) ESPRIT algorithms using second-order statistics and fourth-order cumulants. To avoid ambiguity, we refer to the second-order ESPRIT as ESPRIT2, the higher order ESPRIT as ESPRIT4, and the virtual ESPRIT using fourth-order cumulants as VESPA. The ESPRIT2 and ESPRIT4 algorithms require two subarrays of sensors that are identical but separated by a known displacement vector $\Delta$ with magnitude $\Delta$. The outputs of the subarrays are modeled as

$$x(t) = As(t) + n_x(t)$$

$$y(t) = A\Phi s(t) + n_y(t)$$

where $A$ is the $m \times d$ array response matrix associated with each of the (identical) subarrays, $s(t)$ is the $d \times 1$ vector of
narrowband zero-mean signals, \( n_x(t) \) and \( n_y(t) \) are \( m \times 1 \) Gaussian noise vectors of the \( X \) and \( Y \) subarrays, respectively, and \( \Phi \) is a \( d \times d \) diagonal matrix of phase delays between the doublet sensors for the \( d \) signals. The matrix \( \Phi \) is given by

\[
\Phi = \begin{bmatrix}
    e^{j\phi_1} & 0 & \cdots & 0 \\
    0 & e^{j\phi_2} & \cdots & 0 \\
    \vdots & \vdots & \ddots & \vdots \\
    0 & 0 & \cdots & e^{j\phi_d}
\end{bmatrix}
\]  

where

\[
\phi_k = \frac{\omega_0\Delta \sin \theta_k}{c}
\]

\( \theta_k \) is the direction of arrival of the \( k \)th source, \( \omega_0 \) is the carrier frequency, and \( c \) is the speed of propagation.

Now we combine the subarray outputs to get

\[
z(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \tilde{A}s(t) + n_x(t)
\]

where

\[
\tilde{A} = \begin{bmatrix} A \\ A\Phi \end{bmatrix}
\]

and

\[
n_x(t) = \begin{bmatrix} n_x(t) \\ n_y(t) \end{bmatrix}
\]

A. Second-Order ESPRIT

The least-squares (LS) ESPRIT algorithm using second-order statistics is presented by Roy and Kailath [12]. Forming the covariance matrix of \( z(t) \), we get

\[
R_{zz} = E\{z(t)z^H(t)\} = \tilde{A}R_{ss}\tilde{A}^H + \sigma^2\Sigma_n
\]

where \( R_{ss} \) is the source covariance matrix. The \( 2m - d \) smallest generalized eigenvalues of \( (R_{zz}, \Sigma_n) \) are equal to \( \sigma^2 \). The \( d \) generalized eigenvectors corresponding to the \( d \) largest generalized eigenvalues are used to form

\[
E_s = \Sigma_n[\begin{array}{c} e_1 \\ e_2 \cdots e_d \\ s_1 \\ s_2 \cdots s_d \end{array}]
\]

Because \( \Sigma_n \) must be a known matrix, we simply assume that \( \Sigma_n = I \) in this paper. This assumption also implies that the sensors are not shared between the two subarrays, in which case \( \Sigma_n \) would not be diagonal, even if the noise is white. The case of overlapping subarrays is examined in [8]. However, for fourth-order ESPRIT and VESPA algorithms, the arrays can be overlapping and the noise can be colored, as long as it is Gaussian. We shall call \( s_1, \ldots, s_d \) the signal eigenvectors. The range of \( E_s \) is equal to that of \( \tilde{A} \) [12] (i.e., \( \mathcal{R}(E_s) = \mathcal{R}(\tilde{A}) \)). Thus, there exists a nonsingular matrix \( T \) such that

\[
E_s = \tilde{A}T
\]

Decomposing \( E_s \) into two \( m \times d \) matrices \( E_x \) and \( E_y \), we get

\[
E_s = \begin{bmatrix} E_x \\ E_y \end{bmatrix} = \begin{bmatrix} AT \\ A\Phi T \end{bmatrix}
\]

with

\[
\mathcal{R}(E_x) = \mathcal{R}(E_y) = \mathcal{R}(\tilde{A}).
\]

Since the range of \( E_x \) and the range of \( E_y \) are equal, the rank of the matrix \( [E_x \ E_y] \) is \( d \), which implies that there exists a \( 2d \times d \) matrix

\[
P = \begin{bmatrix} P_x \\ P_y \end{bmatrix}
\]

of rank \( d \), which spans the null-space of \( [E_x \ E_y] \). That is,

\[
0 = [E_x \ E_y]P = E_xP_x + E_yP_y = ATP_x + A\Phi TP_y
\]

or

\[
-ATP_xP_y^{-1} = A\Phi T.
\]

We define a matrix

\[
F = -P_xP_y^{-1}.
\]

From (11), (15), and (16), we see that

\[
E_xF = E_y
\]

or

\[
F = E_x^\#E_y
\]

where \( E_x^\# \) denotes the pseudoinverse of \( E_x \). Combining (15) and (16), we get

\[
ATF = A\Phi T.
\]

Since \( T \) is invertible and \( A \) has full rank, it follows from (19) that

\[
\Phi = TFT^{-1}.
\]

This says that \( F \) and \( \Phi \) are similar and, hence, they have the same eigenvalues, which are the diagonal elements of \( \Phi \). Using (3) and (4), the DOA’s can be computed from the eigenvalues of \( F \).

B. Fourth-Order ESPRIT

In recent years, much attention has been given to signal processing techniques using higher order statistics. One of the main motivations is that a higher order (>2) cumulant is insensitive to additive Gaussian noise regardless of whether it is white or colored. Thus, it is not necessary to model the noise when it is additive Gaussian. For signal processing using higher order statistics, the model used in (1) and (2)
has an extra requirement that the signals in $s(t)$ are non-
Gaussian. Because the signals are assumed to have zero mean,
the fourth-order cumulant is given by

$$\kappa_4(k_1, k_2; l_1, l_2)$$

$$= \sum z_{k_1} z_{k_2} z_{l_1} z_{l_2}$$

$$- E\{z_{k_1} z_{k_2} z_{l_1} z_{l_2}\} - E\{z_{k_1} z_{k_2} z_{l_1} z_{l_2}\}$$

$$- E\{z_{k_1} z_{k_2} z_{l_1} z_{l_2}\} - E\{z_{k_1} z_{k_2} z_{l_1} z_{l_2}\}.$$

(21)

We shall use the notation $\kappa_4(k_1, k_2; l_1, l_2)$, where $k_1, k_2, l_1, l_2$ are the subscripts of the array sensor elements,
to denote the fourth-order cumulant whenever there is no
confusion about which sensor element (of which subarray)
is being referred.

A fourth-order ESPRIT algorithm has been proposed by
Chiang and Nikias [2] for the case of equispaced sensors and
statistically independent sources. The steering matrix is given by

$$A = \begin{bmatrix}
1 & e^{j\omega_1} & \ldots & e^{j\omega_d} \\
1 & e^{j\omega_2} & \ldots & e^{j\omega_d} \\
\vdots & \vdots & \ddots & \vdots \\
e^{j\omega_{(m-1)}} & e^{j\omega_{(m-2)}} & \ldots & e^{j\omega_{(m-1)}}
\end{bmatrix}.$$ (22)

In the second-order ESPRIT algorithm, it is not necessary to
know the form of the steering matrix, only that there are two
identical arrays and that the displacement vector $\Delta$ is known.
However, for the fourth-order ESPRIT algorithm, when the
steering matrix is of the form (22), then the cumulant matrices
(23) and (24) (shown at the bottom of the page) have the forms

$$C_{xx} = \Gamma A^H \Gamma A$$

$$C_{xy} = \Gamma A^H \Gamma$$

and $\Gamma$ is the matrix

$$\Gamma = \begin{bmatrix}
\gamma_1 & 0 & \ldots & 0 \\
0 & \gamma_2 & 0 & \ldots \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & \gamma_d
\end{bmatrix}.$$ (27)

where $\gamma_i$ is the kurtosis of the $i$th source. In the above
equations, $c(x_i, x_j, x_k, x_l)$ represents the cumulant of the
arguments $x_i, x_j, x_k, x_l$. The matrix $C_{xx}$ in (23) is not
identical to the one proposed in [2], although they are equivalent
[i.e., they both have the equality in (25)]. In the case of finite
data, the sample estimate of the matrix in (23) is Hermitian
while the one proposed in [2] is not.

In a manner similar to the second-order ESPRIT, we can
formulate a combined output as in (5)-(7). Doing this, we can
get the fourth-order cumulant matrix

$$C_{zz} = \overline{A} \Gamma A^H$$

$$= \begin{bmatrix}
C_{xx} & C_{xy} \\
C_{xy}^H & C_{xx}
\end{bmatrix}.$$ (28)

From this point, the fourth-order ESPRIT algorithm proceeds
as in the second-order case, with the covariance matrix $R_{zz}$
replaced by the cumulant matrix $C_{zz}$ in (8).

C. Virtual ESPRIT

Recently, Dogan and Mendel [3] have given a novel interpre-
tation to cumulants for narrowband array processing. Bas-
ically, they have shown that, for independent sources, a fourth-
order cross-cumulant can be interpreted as a (second-order)
cross-covariance. The interesting insight of this interpretation
is that for some cumulants, the interpreted cross-covariance is
a “virtual” cross-covariance. That is, some cross-covariances
are between data received at real sensors and “virtual” sensors
(i.e., sensors that don’t physically exist). Dogan and Mendel
have applied the idea of virtual sensors to several array
processing problems, such as implementing a virtual ESPRIT
algorithm (VESPA) and suppressing non-Gaussian additive
noise. Here, we give a brief summary of the interpretation.

Single Source: First assume that we have a single (non-
Gaussian) source with a propagation vector $\bar{k}$, where $\bar{k} = k_x \bar{a}_x + k_y \bar{a}_y$ ($\bar{a}_x$ and $\bar{a}_y$ are unit vectors along the $x$ and $y$
axis, respectively), with variance $\sigma_i^2$ and kurtosis $\gamma_i$.

Referring to Fig. 1, the shaded circles denote real sensors
and the clear circles denote a virtual sensor. Let the reference
doubt be $r_1$ and $r_1$ (both real sensors), then the signals
received at the real sensors are, for example

$$r_1(t) = a_1 s(t)$$

$$r_2(t) = a_2 s(t) \exp\{ -j(\bar{k} \cdot \bar{d}_{12}) \}$$

$$v_1(t) = a_1 s(t) \exp\{ -j(\bar{k} \cdot \bar{\Delta}) \}$$

(29)

and at the virtual sensor

$$v_2(t) = a_2 s(t) \exp\{ -j(\bar{k} \cdot \bar{d}_{12}) \}$$

(30)
where $a_i$ is the response (i.e., gain and phase) of the $i$th sensor. The noiseless cross-covariance between $r_k(t)$ and $v_l(t)$ is

$$E[r_k^*(t)v_l(t)] = E[a_k^*s^*(t) \exp[j(k \cdot d_{1k})]\cdot a_l(t) \exp[-j(k \cdot d_{1l})]$$

$$= a_k^*a_l^2 \exp[j(k \cdot d_{1k})] \exp[-j(k \cdot d_{1l})].$$

Now consider the fourth-order cross-cumulant

$$\text{Cum} [r_k^*(t), v_l(t), r_k^*(t), v_l(t)]$$

$$= \text{Cum} \{a_k^*s^*(t), a_l(t) \exp[-j(k \cdot d_{1l})],$$

$$a_k^*s^*(t) \exp[j(k \cdot d_{1k}), a_l(t) \exp[-j(k \cdot d_{1l})]\}$$

$$= |a_k|^2 a_k^*a_l^2 \gamma_{11} \exp[j(k \cdot d_{1k})] \exp[-j(k \cdot d_{1l})].$$

Comparing (31) and (32), we have

$$E[r_k^*(t)v_l(t)] = \frac{\sigma^2}{|a_k|^2 \gamma_{11}} \text{Cum} [r_k^*(t), v_l(t), r_k^*(t), v_l(t)].$$

This equation shows that a fourth-order cumulant of the output from real physical sensors is proportional to a cross-covariance of output from a real and a virtual sensor. In other words, we have an interpreted array that has a larger aperture than the physical array.

**Multiple Sources:** Assume now that we have two statistically independent (non-Gaussian) sources impinging on an array. In this case, the noiseless cross-covariance between $r(t)$ and $v(t)$ is

$$E[r_k^*(t)v_l(t)]$$

$$= E\{a_k^*s_1^*(t) \exp[j(k_1 \cdot d_{1k})],$$

$$+ a_k^*a_l^2 \sigma_2^2 \exp[j(k_2 \cdot d_{1k})]\cdot \exp[-j(k_1 \cdot d_{1l})]$$

$$= a_k^*a_l^2 a_1^2 \sigma_2^2 \exp[j(k_1 \cdot d_{1k})] \exp[-j(k_2 \cdot d_{1k})]$$

$$= a_k^*a_l^2 \sigma_2^2 \exp[j(k_1 \cdot d_{1k})] \exp[-j(k_2 \cdot d_{1k})]$$

$$\cdot \exp[-j(k_1 \cdot d_{1l})] + a_k^*a_l^2 \sigma_2^2 \exp[j(k_2 \cdot d_{1k})]$$

$$\cdot \exp[-j(k_2 \cdot d_{1l})].$$

(34)

where $a_{ij}$ is the response of the $i$th sensor to the $j$th source. Now consider the fourth-order cross-cumulant

$$\text{Cum} [r_k^*(t), v_l(t), r_k^*(t), v_l(t)]$$

$$= \text{Cum} \{a_k^*s_1^*(t) + a_k^*a_l^2 \gamma_{11},$$

$$a_1s_1(t) \exp[-j(k_1 \cdot d_{1k})],$$

$$a_1s_1(t) \exp[-j(k_1 \cdot d_{1l})]\}$$

$$= \gamma_{11} a_k^2 a_l^4 \gamma_{11} \exp[-j(k_1 \cdot d_{1k})]$$

$$\cdot \exp[j(k_1 \cdot d_{1k})] \exp[-j(k_1 \cdot d_{1l})]$$

$$+ \gamma_{11} a_k^2 a_l^4 \gamma_{11} \exp[-j(k_1 \cdot d_{1k})]$$

$$\cdot \exp[j(k_1 \cdot d_{1k})] \exp[-j(k_1 \cdot d_{1l})].$$

(35)

Comparing (34) and (35), we see that the cumulant and covariance are no longer proportional as is the case in (33). However, we can still interpret (35) as a covariance with sources from the same directions. The only difference is that the sources have different “power” levels. However, the DOA information is not altered. Note that $\gamma_{11}$, being a cumulant, may be negative, whereas a true power is always positive.

This case is representative of the case of multiple sources because of the cumulant property that if the random variables $x_1, x_2, \ldots, x_n$ are independent of the random variables $y_1, y_2, \ldots, y_n$, then

$$\text{Cum} [x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n] =$$

$$\text{Cum} [x_1, x_2, \ldots, x_n] + \text{Cum} [y_1, y_2, \ldots, y_n].$$

(36)

This means that for several sources, the cumulant of the sum of sources is simply the sum of the cumulant of each source.

Given an array of sensors and one additional sensor, we can have, using this interpretation, a virtual copy of the first array, allowing us to implement the VESPA algorithm. The additional sensor has identical characteristics to the reference sensor and the pair forms a doublet. For example, if we have an array of four sensors and one additional sensor (five total), the cumulant matrix used to implement the least-squares VESPA
in this case would be
\[
V_{zz} = \begin{bmatrix}
V_{xx} & V_{yx}^T \\
V_{yx} & V_{yy}
\end{bmatrix}
\]  

where we obtain (38) and (39) (shown at the bottom of the page), where \( c(x_1^*, x_2, x_3^*, x_4) \) represents the cumulant of the arguments \( x_1^*, x_2, x_3^*, x_4 \). It is shown in [3] that, with five physical sensors, VESPA can be implemented with two subarrays of five sensors in each subarray instead of only four sensors in each subarray. In this case, \( V_{xx} \) and \( V_{yx} \) would be \( 5 \times 5 \) matrices instead of the \( 4 \times 4 \) matrices shown in (38) and (39). This larger virtual subarray is used for the examples in Section IV. The VESPA algorithm now proceeds as in the second-order case, with \( R_{zz} \) replaced by \( V_{zz} \) in (8).

From a different standpoint, we can view VESPA as using the available cumulants in a more efficient way. For instance, in an \( m \)-element array of sensors, there are \( m^4 \) possible cumulants. ESPRIT4 requires two \( m \)-element arrays, which means there are \( 16m^4 \) possible cumulants. ESPRIT4 uses only \( 4m^2 \) out of \( 16m^4 \) possible cumulants. VESPA, on the other hand, uses \( 4m^2 \) out of \( m^4 \) possible cumulants.

### III. Asymptotic Performance Analysis of ESPRIT

In this section, we derive the asymptotic variance of the estimated DOA’s. The derivation in this paper is along the lines of the first-order analysis done by Rao and Hari [11]. In that paper, Rao and Hari derived the asymptotic variance of the estimated DOA’s for the second-order LS-ESPRIT algorithm under the assumption of Gaussian signals. Furthermore, they showed that the asymptotic variance of LS-ESPRIT is equal to that of the total LS-ESPRIT algorithm, whether the signals are Gaussian or not. The assumption of this paper is that the signals may be non-Gaussian and, hence, the variance expressions of [11] in general do not apply. However, the methodology of [11] is followed here.

In LS-ESPRIT, we compute a matrix \( F \) using (18) and then perform an eigendecomposition to get its associated eigenvalues. Let \( \lambda_i \) be an eigenvalue of \( F \), \( v_i \) be the corresponding eigenvector, and \( q_i \) be the corresponding left eigenvector, such that
\[
Fv_i = \lambda_i v_i, \\
q_i^TF = \lambda_i q_i.
\]  

Furthermore, the left and right eigenvectors are orthogonal [13] and we can normalize the left eigenvectors so that
\[
q_i v_i = 1.
\]

Equations (40) and (41) imply that
\[
\lambda_i = q_i F v_i.
\]

Under most circumstances, the matrix \( F \) has to be estimated using finite data, so that, from (18)
\[
\hat{F} = \hat{E}_x^T \hat{E}_y.
\]

An error \( \delta F = \hat{F} - F \) in estimating \( F \) will cause an error \( \delta \lambda_i = \hat{\lambda}_i - \lambda_i \). As a first-order approximation, the error \( \delta \lambda \) can be written as
\[
\delta \lambda_i \approx q_i \delta F v_i.
\]

It follows from (18) that we have the first-order approximations
\[
(E_c + \delta E_c)(F + \delta F) \approx E_y + \delta E_y, \\
E_x \delta F \approx \delta E_x F
\]

and
\[
\delta F \approx E_x^T \delta E_y - E_x^T \delta E_c F.
\]

Using (47) in (44) and noting that \( |\lambda_i|^2 = 1, \) we get
\[
\delta \lambda_i \approx q_i E_x^T (\delta E_y v_i - E_c F v_i) \\
= q_i E_x^T (\delta E_y v_i - \lambda_i \delta E_x v_i) \\
= q_i E_x^T (W_1 - \lambda_i W_2) \delta E_c v_i \\
= -\lambda_i q_i E_x^T (W_1 - \lambda_i^* W_2) \delta E_c v_i
\]

where
\[
W_1 = [I_m \ 0_m] \quad (49) \\
W_2 = [0_m \ I_m] \quad (50)
\]

and
\[
\delta E_c = \hat{E}_c - E_x. 
\]

The matrix \( E_x \) is defined in (9), \( I_m \) is the \( m \times m \) identity matrix, and \( 0_m \) is the \( m \times m \) zero matrix. In the following
\begin{equation}
E\{\delta \lambda_i \}^2 = \eta_i E_x^H \{W_1 - \lambda_i^* W_2\} E\{\delta E_x \delta E_x^H\} \cdot E\{W_1 - \lambda_i^* W_2\}^H (q_i E_x^H)^H E\{\delta \lambda_i \} \cdot \delta \lambda_i \, \delta \lambda_i^* .
\end{equation}

(59)

This expression is common to all three algorithms considered in this paper. The difference among the three ESPRIT algorithms is in the statistic used for estimation. ESPRIT2 uses a covariance matrix $R_{zz}$; ESPRIT4 uses a fourth-order cumulant matrix $C_{zz}$; VESPA uses a fourth-order cumulant matrix $V_{zz}$. Each of these matrices produces a set of eigenvectors $\{s_1, \ldots, s_k\}$ that has different statistical properties from the eigenvectors produced by the other matrices. In other words, the covariances of the eigenvectors, $E\{\delta a_i \delta b_j^H\}$ and $E\{\delta s_g \delta s_h^T\}$, which appear in (52) and (53), are unique to each algorithm. Therefore, in the following subsections, we derive these covariances for each algorithm.

A. Covariance of Eigenvectors—ESPRIT2

The first-order Taylor series expansion of the eigenvectors (corresponding to distinct eigenvalues) of a $2m \times 2m$ matrix $R_{zz}$ [14] is

\begin{equation}
\hat{s}_g = s_g + \sum_{i=1}^{2m} \sum_{a_1=1}^{2m} \sum_{a_1=1}^{2m} \frac{s_{g_i}}{\alpha_g - \alpha_i},
\end{equation}

(60)

where $\alpha_i$ is an eigenvalue of the matrix $R_{zz}$ and $s_{g_i}$ is the $i$th element of the eigenvector $s_g$. The notation $(\cdot)_i$ refers to the $(m, n)$ element of a matrix. The expansion in (60) assumes that the matrix $(R_{zz} - R_{zz}^*)$ is conjugate symmetric and that the eigenvectors $s_{1}, s_{2}, \ldots, s_{2m}$ are orthonormal. That is

\begin{equation}
s_{i} s_{j} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}.
\end{equation}

(61)

When $s_i$ corresponds to a simple eigenvalue, the eigenvector $s_i$ is uniquely determined up to a complex constant. When there are repeated eigenvalues, the corresponding eigenvectors are not uniquely determined. This will be the case for the eigenvectors $s_{d+1}, \ldots, s_{2m}$, which correspond to a single eigenvalue of multiplicity $2m - d$. Also, all the eigenvectors have to be normalized to have unity norm. It follows from (60) that the covariance matrix of the signal eigenvectors $s_1$ and $s_k$ is, to the first order, shown in (62), at the bottom of the next page. We note that this expression is greatly simplified when the signals are Gaussian, as is assumed in [11]. The covariance of sample covariances, needed to compute (62) is derived in Appendix B, (88), and repeated here for convenience.

\begin{equation}
E\{\delta R_{zz} \} = \frac{1}{N_s} \mu_2(a_1, b_1; a_2, b_2) - \frac{1}{N_s} \mu_2(a_1; a_2) \cdot \mu_2(b_1; b_2).
\end{equation}

(63)

The second-order and fourth-order moments, denoted by $\mu_2(\cdot)$ and $\mu_4(\cdot)$, respectively, of the received data $z(t)$ are given by

\begin{equation}
\mu_2(m; n) = E\{z_m(t) z_n^*(t)\}
\end{equation}

(64)
and

$$\mu_4(m_1, m_2; n_1, n_2) = E\{z_{m_1}(t)z_{m_2}(t)z_{m_1}(t)z_{m_2}(t)\}. \quad (65)$$

Furthermore, the unconjugated covariance of the sample eigenvectors \(s_g\) and \(s_h\) is, to the first order

$$E\{s_g s_h^T\} = \sum_{i=1}^{2m} \sum_{j=1}^{2m} \left( s_{i+a}^* s_{j+b} s_{i-a} s_{j-b} \right)$$

$$\cdot \frac{E\{(|R_{zz} - R_{zz})_{a1,a2} (R_{zz} - R_{zz})_{b1,b2}\}}{s_g s_h^T} \cdot (\alpha_g - \alpha_l)(\alpha_h - \alpha_n). \quad (66)$$

Because of the inherent symmetry in the covariance matrix \(R_{zz}\), (63) can be used to compute \(E\{(|R_{zz} - R_{zz})_{a1,a2} (R_{zz} - R_{zz})_{b1,b2}\}\) by swapping the indices \(a_1\) and \(b_1\).

B. Covariance of Eigenvectors—ESPRIT4

Since (60) is valid for eigenvectors of any Hermitian matrix, we can use it as an estimate of the eigenvectors of a cumulant matrix. We simply replace \(R_{zz}\) with \(C_{zz}\) and \(R_{zz}\) with \(C_{zz}\).

The covariance matrix of the signal eigenvectors \(s_g\) and \(s_h\) for a cumulant matrix, to the first order, is shown in (67) at the bottom of the next page. The sample cumulant of (21) is given by

$$\hat{k}_4(a_1, a_2; b_1, b_2) = \left\langle C_{zz} \right\rangle_{a1,a2}$$

$$= \frac{1}{N} \sum_{r=1}^{N} z_{k1}(t)z_{k1}^*(t)z_{k2}(t)z_{k2}^*(t)$$

$$- \frac{1}{N^2} \sum_{r=1}^{N} z_{k1}(t)z_{k1}^*(t) \sum_{s=1}^{N} z_{k2}(s)z_{k2}^*(s)$$

$$- \frac{1}{N^2} \sum_{r=1}^{N} z_{k2}(t)z_{k2}^*(t) \sum_{s=1}^{N} z_{k1}(s)z_{k1}^*(s)$$

$$- \frac{1}{N^2} \sum_{r=1}^{N} z_{k1}(t)z_{k2}(t) \sum_{s=1}^{N} z_{k1}^*(s)z_{k2}^*(s)$$

$$+ \frac{1}{N^2} \sum_{r=1}^{N} z_{k1}(t)z_{k2}(t) \sum_{s=1}^{N} z_{k1}^*(t)z_{k2}^*(s)$$

and its asymptotic covariance is given in (81) of the Appendix.

That is, (81) is used to compute \(E\{(|C_{zz} - C_{zz})_{a1,a2} (C_{zz} - C_{zz})_{b1,b2}\}\) in (67). Note that the indices \((a_1, a_2)\), which refer to the row and column of a matrix, do not in general have a direct mapping to the indices \((k_1, k_2, l_1, l_2)\). The exact mapping depends on which cumulants are used to form the matrix \(C_{zz}\) and how the cumulants are ordered within \(C_{zz}\).

The same comment applies to \((b_1, b_2)\) and \((m_1, m_2, n_1, n_2)\).

Furthermore, the unconjugated covariance of the sample eigenvectors \(s_g\) and \(s_h\), to the first-order, is

$$E\{s_g s_h^T\} = \sum_{i=1}^{2m} \sum_{j=1}^{2m} \left( s_{i,a}^* s_{j,b} s_{i,a} s_{j,b} \right)$$

$$\cdot \frac{E\{(|C_{zz} - C_{zz})_{a1,a2} (C_{zz} - C_{zz})_{b1,b2}\}}{s_g s_h^T} \cdot (\alpha_g - \alpha_l)(\alpha_h - \alpha_n). \quad (69)$$

Because of the inherent symmetry in the fourth-order cumulant, (81) can also be used to compute \(E\{(|C_{zz} - C_{zz})_{a1,a2} (C_{zz} - C_{zz})_{b1,b2}\}\) by swapping the indices \((m_1, m_2)\) with \((n_1, n_2)\).

C. Covariance of Eigenvectors—VESPA

From a mathematical point of view, there is very little difference between the performance equations used for the ESPRIT4 and VESPA algorithms. The only difference is that the cumulants used for the two algorithms are different. Therefore, the covariance of sample cumulants

$$E\{\hat{C}_{zz} - C_{zz}\}_{a1,a2} (\hat{C}_{zz} - C_{zz})_{b1,b2}^*\} \quad (70)$$

in (67)—and the unconjugated counterpart in (69)—will involve different cumulants, which are given in the matrix \(V_{zz}\) in (37). Therefore, the conjugated and unconjugated covariances of the eigenvectors for VESPA are given by

$$E\{s_g s_h^T\} = \sum_{i=1}^{2m} \sum_{j=1}^{2m} \left( s_{i,a}^* s_{j,b} s_{i,a} s_{j,b} \right)$$

$$\cdot \frac{E\{(|V_{zz} - V_{zz})_{a1,a2} (V_{zz} - V_{zz})_{b1,b2}\}}{s_g s_h^T} \cdot (\alpha_g - \alpha_l)(\alpha_h - \alpha_n). \quad (71)$$

and

$$E\{s_g s_h^T\} = \left\{ \begin{array}{l}
E\left[ \sum_{i=1}^{2m} \sum_{j=1}^{2m} \left( s_{i,a}^* s_{j,b} s_{i,a} s_{j,b} \right) \right]
\end{array} \right\}$$

$$\cdot \frac{E\{(|R_{zz} - R_{zz})_{a1,a2} (R_{zz} - R_{zz})_{b1,b2}\}}{s_g s_h^T} \cdot (\alpha_g - \alpha_l)(\alpha_h - \alpha_n). \quad (62)$$
and
\[
E\{s_g s_h^T\} = \sum_{i: z} \sum_{h} s_{1j} a_1 s_{2j} a_2 s_{3k} b_1 s_{4k} b_2 \cdot E\{(V_{zz} - V_{zz})_{a1a2}(V_{zx} - V_{zx})b_1b_2\} \cdot \frac{s_t s_m^T}{(\alpha_g - \alpha_l)(\alpha_h - \alpha_n)}.
\]  

Note that \(m\) for VESPA is the total number of sensors and is one more than \(m\) for ESPRIT4 or ESPRIT2, where \(m\) is the number of sensors in each subarray. We note that the covariances in (71) and (72) can still be computed using (81). From a practical point of view, the advantage of VESPA is that the cumulants can be computed from data collected from fewer sensors.

IV. SIMULATION AND NUMERICAL EVALUATION

In this section, we examine several cases of interest and evaluate the asymptotic results derived in the previous sections. In addition, we use Monte Carlo simulations to verify the theoretical results for these cases. In all cases, we assume that the number of sources is perfectly known.

We note that the theoretical results for ESPRIT4 and VESPA in Section III are valid for spatially colored (Gaussian) noise, while the result for ESPRIT2 is valid for spatially white noise only. It is straightforward to extend the results of ESPRIT2 to the colored noise case where the covariance matrix of the noise is known. However, this is not done in the previous analysis because ESPRIT4 and VESPA do not require the noise covariance matrix to be known. To make a reasonable comparison of the three algorithms, we will use in all the examples a white Gaussian noise scenario. Furthermore, as noted in Section II-A, the analysis presented for ESPRIT2 must be modified when overlapping subarrays are used. However, the analysis for ESPRIT4 and VESPA are valid even when the subarrays are overlapping, given that the noise is Gaussian.

Example 1—Standard Deviation Versus DOA Separation: In this example, we let two equal power statistically independent 4-QAM sources impinge on two uniformly spaced linear subarrays. Each subarray has four omnidirectional sensors. Each sensor is spaced 0.5 wavelengths apart and the subarrays are spaced 0.5 wavelengths apart. The subarrays are depicted in Fig. 2, where the angle-of-arrival of a single source is shown. As explained in [3], when given an \(M\) element subarray and one guiding sensor, the resulting subarrays used for VESPA have \(M + 1\) elements. The steering vector for subarray 1 (excluding \(r_0\)) has the form
\[
a(\theta) = \begin{bmatrix}
1 \\
e^{-j2\pi \cos \theta} \\
e^{-j3\pi \cos \theta}
\end{bmatrix}.
\]

The ESPRIT2, ESPRIT4, and VESPA algorithms are used for DOA estimation. There are a total of eight sensors for ESPRIT4 and ESPRIT2. We use two versions of the VESPA algorithms, which we denote by VESPA5 and VESPA8. For VESPAs, we use a four-element linear array and one more sensor as the guiding sensor (i.e., five sensors total). To make a comparison of the algorithms using the same number of real sensors, we use the array shown in Fig. 3 for VESPA8. The DOA of source 2 is fixed at 40° while the DOA of source 1 is varied from 0° to 180°. The signals are corrupted by additive white Gaussian noise with input signal-to-noise ratio (ISNR) of 10 dB. The results are shown in Figs. 4 and 5. The lines are theoretical values while the discrete points are simulation values. We use 500 independent runs with 5000 snapshots per run. The simulation and theoretical results show good agreement. Notice, in the figures, the standard deviation increases as the DOA of source 1 approaches the DOA of source 2 (i.e., 40°). Furthermore, the standard deviation for source 1 increases as source 1 approaches 90°. In the region of small DOA separation between the two sources, ESPRIT2 and VESPA5 are comparable in performance. It is somewhat surprising that in this region, VESPA8 has slightly worse performance than VESPA5. However, because the two versions of VESPA and ESPRIT4 are implemented with different cumulants, the results demonstrate that some cumulants may provide more information. For larger DOA separation or when the DOA is near 90°, ESPRIT2 performs the best.
Fig. 2. Subarrays of sensors used in Examples 1 and 2. For ESPRIT2 and ESPRIT4, subarray 1 consists of $r_1$ and $r_2$, while subarray 2 consists of $v_1$ and $v_2$, and all circles represent real sensors except for $v_5$, which is a virtual sensor used only in VESPA5. For VESPA5, subarray 1 consists of $r_1$ and $r_3$, while subarray 2 consists of $v_1$ and $v_2$, and the shaded circles represent real sensors while the clear circles represent virtual sensors. A source approaches the array at an angle $\theta$.

**Example 2—Standard Deviation Versus ISNR:** We repeat the experiment using the parameters in Example 1, except that we vary the input SNR level, and the simulations are done with 500 snapshots. The DOA's of the two sources are fixed. Source 1 impinges on the array at 20° and source 2 impinges on the array at 40°. The results for source 1 are shown in Fig. 6. The results for source 2 look very similar and, hence, we have not included them in this paper. The simulation and theoretical results show good agreement, even at 500 snapshots. In this example, the difference in performance between VESPA5 and VESPA8 is more clearly shown. As in Example 1, ESPRIT2 again has the best performance.

**Example 3—Standard Deviation Versus DOA Separation for T-Array:** In this example, we apply the same signal parameters as Example 1 to an array used in [3]. The array configuration is shown in Fig. 7. We shall refer to this configuration as the (inverted) T-array. Furthermore, to make a comparison for VESPA using the same number of sensors as ESPRIT2, we also use the configuration in Fig. 8. As in previous examples, we call the resulting algorithm VESPA8. Because the ESPRIT4 algorithm in [2] is formulated only for linear arrays, we compute the results only for the ESPRIT2, VESPA5, and VESPA8 algorithms.

Fig. 3. Subarrays of sensors used in Examples 1 to 2 for VESPA8. Subarray 1 consists of $r_1$ and $r_5$, while subarray 2 consists of $v_1$ and $v_6$. The shaded circles represent real sensors while the clear circles represent virtual sensors. There is an overlap of four real sensors in subarray 2. A source approaches the array at an angle $\theta$.

Fig. 4. Results for Example 1. The standard deviation of estimated DOA for source 1, plotted as a function of the DOA of source 1 for two linear subarrays with four sensors each. The DOA of source 2 is fixed at 40°. The dash and solid lines are the theoretically predicted results and the discrete points are simulation results. Five thousand snapshots are used.

The response of each sensor takes the form $-\cos(\theta)$ and if it is rotated clockwise by $\alpha$, then its response is $\cos(\theta + \alpha)$. The sensors used in this example have orientations $\{0°, \pm 5°, \pm 10°\}$ with locations $\{(0, 0), (0, -\lambda/2), (-\lambda/2, -\lambda), (\lambda/2, -\lambda)\}$ on the $x-y$
plane. Thus, a steering vector has the form\(^1\)

\[
a(\theta) = \begin{bmatrix}
\cos(\theta) \\
\cos(\theta - 5^\circ) e^{-j\pi \cos \theta} \\
\cos(\theta + 3^\circ) e^{-j\pi (2 \cos \theta + \sin \theta)} \\
\cos(\theta - 10^\circ) e^{-j\pi (2 \cos \theta - \sin \theta)}
\end{bmatrix}
\]

(74)

(The frame of reference for \(\theta\) used in this paper is not the same as in [3], so that the sensor response looks analytically different from that given in [3].) We plot the standard deviation for both sources in Figs. 9 and 10. The dash and solid lines are the theoretically predicted results and the discrete points are simulation results.

\(^1\)Equation (74) is for ESPRIT2 only. The steering vector has to be modified slightly for VESPA5 and VESPA8 to reflect the larger number of sensors for each subarray.

Fig. 5. Results for Example 1. The standard deviation of estimated DOA for source 2, plotted as a function of the DOA of source 1 for two linear subarrays with four sensors each. The DOA of source 2 is fixed at 40\(^\circ\). The dash and solid lines are the theoretically predicted results and the discrete points are simulation results. Five thousand snapshots are used.

Fig. 6. Results for Example 2. The standard deviation of estimated DOA for source 1, plotted as a function of the input SNR. Source 1 and 2 impinge on the array at 20\(^\circ\) and 40\(^\circ\), respectively. Two linear arrays with four sensors each. 500 snapshots.

Fig. 7. Array configuration used in [3]. Configuration in (a) is used for VESPA5 and (b) is used for ESPRIT2. The shaded triangles and X’s represent real sensors, and the clear triangles represent virtual sensors.

Fig. 8. Array configuration used for VESPA8 in [3]. Five sensor elements are common to both subarrays. The clear triangles represent virtual sensors.

Example 4—Standard Deviation Versus ISNR for T-Array:
We repeat the experiment using the parameters in Example 3, except that we vary the input SNR level and the simulations are done with 500 snapshots. The DOA’s of the two sources are fixed. Source 1 impinges on the array at 20\(^\circ\) and source 2 impinges on the array at 40\(^\circ\). The results for source 1 are shown in Fig. 11. The results for source 2 look very similar and, hence, we have not included them in this paper. The dash and solid lines are the theoretically predicted results and the discrete points are simulation results.

It is clear that in these examples, where the noise is additive white Gaussian and the signals are 4-QAM, the second-order ESPRIT algorithm performs the best among the three
algorithms. We emphasize that when the noise is colored with unknown covariance matrix, it is better to use the higher order methods. It is interesting that the VESPA5 algorithm performs just as well as ESPRIT2 and better than ESPRIT4 in the region of small DOA separation, even though the latter algorithms require more sensors. While the performance of VESPA8 in most cases is better than VESPA5, VESPA8 still trails ESPRIT2. Because ESPRIT4, VESPA5, and VESPA8 use different cumulants to form their respective cumulant matrices, the difference in performance suggests that using a different set of cumulants will improve or degrade the performance of an algorithm. However, it is unclear which set of cumulants would yield the best performance.

It is noteworthy that the simulation results in [3] show that the performance of VESPA exceeds that of ESPRIT2 for the case of BPSK signals in white Gaussian noise. It is possible that BPSK signals, which takes on fewer possible values than a 4-QAM signal, can be viewed as being “less Gaussian” than 4-QAM signals. Hence, BPSK signals may yield better results for the higher order techniques, which require non-Gaussian signals.

V. CONCLUSION

In this paper, we derived expressions for the asymptotic variance of DOA estimates for three LS-ESPRIT algorithms under a non-Gaussian signal assumption. The formulas derived are used to evaluate the performance of the second-order ESPRIT algorithm, the fourth-order ESPRIT algorithm, and the VESPA algorithm. Monte Carlo simulations are presented to verify the theoretically predicted performance. From the examples we used, where the additive noise is white Gaussian, it appears that the second-order ESPRIT has the best performance.

APPENDIX A

COVARIANCE OF FOURTH-ORDER SAMPLE CUMULANTS

In this appendix, we derive approximate expressions for the covariance of fourth-order sample cumulants. The expressions for zero-mean and symmetrically distributed signals are derived in [10]. We extend that derivation to zero-mean and possibly nonsymmetrically distributed signals.

The fourth-order cumulant and sample cumulant are given by

\[ \kappa_4(k_1, k_2; l_1, l_2) = \mu_4(k_1, k_2; l_1, l_2) - \mu_2(k_1; l_1)\mu_2(k_2; l_2) - \mu_2(k_1; l_2)\mu_2(k_2; l_1) - \mu_2(k_1, k_2)\mu_2(l_1, l_2) \]  

(75)
and
\[
\hat{\kappa}_4(k_1, k_2; l_1, l_2) = \mu_4(k_1, k_2; l_1, l_2) - \mu_2(k_1, k_2; l_1)\mu_2(k_2; l_2) - \mu_2(k_1, k_2; l_2)\mu_2(k_2; l_1) - \hat{\mu}_2(k_1, k_2)\hat{\mu}_2(l_1, l_2)
\]  
(76)
respectively, and \(\mu_m\) denotes the \(m\)th order moment; for example
\[
\mu_4(k_1, k_2; l_1, l_2) = E\{z_{k_1}, z_{k_2}, z_{l_1}, z_{l_2}\}
\]  
(77)
The estimation error of the sample cumulant is
\[
\delta\kappa_4(k_1, k_2; l_1, l_2) = \kappa_4(k_1, k_2; l_1, l_2) - \kappa_4(k_1, k_2; l_1, l_2)
\]  
(78)
which we can approximate, for large \(N_s\), by
\[
\delta\kappa_4(k_1, k_2; l_1, l_2) \approx \delta\mu_4(k_1, k_2; l_1, l_2)
\]  
(79)
Hence
\[
E\{\delta\kappa_4(k_1, k_2; l_1, l_2)\delta\kappa_4(m_1, m_2; n_1, n_2)\}
\]  
\[
\approx E\left[ \begin{bmatrix} \delta\mu_4(k_1, k_2; l_1, l_2) \\
- \sum_{i=1}^{2} \sum_{j=1}^{2} \mu_2(k_{3-i}; l_{3-j})\delta\mu_2(k_i; l_j) \\
- \mu_2(k_1, k_2)\delta\mu_2(l_1, l_2) \\
- \mu_2(l_1, l_2)\delta\mu_2(k_1, k_2) \end{bmatrix}
\right]
\]  
\[
= E\{\delta\mu_4(k_1, k_2; l_1, l_2)\delta\mu_4(m_1, m_2; n_1, n_2)\}
\]  
\[
= \sum_{p=1}^{2} \sum_{q=1}^{2} \mu_2(m_{3-p}; n_{3-q})\delta\mu_2(m_p; n_q) - \mu_2(m_1, m_2)\delta\mu_2(n_1, n_2) - \mu_2(n_1, n_2)\delta\mu_2(m_1, m_2)
\]  
(80)
To complete the derivation, we need the expressions for the covariance of even-order moments. However, for the sake of continuity, the covariance of even-order moments will be derived in the next section. Substituting (87) into (80), we get
\[
N_s \cdot E\{\delta\kappa_4(k_1, k_2; l_1, l_2)\delta\kappa_4(m_1, m_2; n_1, n_2)\}
\]  
\[
\approx \mu_4(k_1, k_2; l_1, l_2)\mu_4(m_1, m_2; n_1, n_2)
\]  
\[
- \sum_{p=1}^{2} \sum_{q=1}^{2} \mu_2(m_{3-p}; n_{3-q})\delta\mu_2(m_p; n_q) - \mu_2(m_1, m_2)\mu_2(n_1, n_2) - \mu_2(n_1, n_2)\mu_2(m_1, m_2)
\]  
\[
- \mu_2(k_1, k_2)\mu_2(l_1, l_2)\mu_2(m_1, m_2)\]
\[
- \mu_2(l_1, l_2)\mu_2(k_1, k_2)\mu_2(m_1, m_2)
\]  
(81)
\[ + \sum_{i=1}^{2} \sum_{j=1}^{2} \sum_{p=1}^{2} \mu_2(k_{3-i}; l_{3-j}) \mu_2(m_{3-p}; n_{3-q}) \]
\[ \cdot \mu_4(k_i, m_p; l_j, n_q) \]
\[ - \mu_2(k_i; l_j) \mu_2(m_p; n_q) \]
\[ + \mu_2(m_{l_1}, m_{l_2}) \sum_{i=1}^{2} \sum_{j=1}^{2} \mu_2(k_{3-i}; l_{3-j}) \]
\[ \cdot \mu_4(k_i; l_j, n_1, n_2) \]
\[ - \mu_2(k_i; l_j) \mu_2(n_1, n_2) \]
\[ + \mu_2(n_1, n_2) \sum_{i=1}^{2} \sum_{j=1}^{2} \mu_2(k_{3-i}; l_{3-j}) \]
\[ \cdot \mu_4(k_i; l_j, n_1, n_2) \]
\[ - \mu_2(k_i; l_j) \mu_2(n_1, n_2) \]
\[ \cdot \mu_4(k_i, m_j, m_2; l_j) \]
\[ - \mu_2(k_i; l_j) \mu_2(m_1, m_2) \]
\[ - \mu_2(k_1, k_2) [\mu_6(m_1, m_2; n_1, n_2, l_1, l_2) \]
\[ - \mu_2(m_1, m_2; n_1, n_2, l_1, l_2) \]
\[ + \mu_2(m_1, m_2; n_1, n_2) \mu_2(l_1, l_2) \]
\[ + \mu_2(k_1, k_2) \sum_{p=1}^{2} \sum_{q=1}^{2} \mu_2(m_3-p; n_3-q) \]
\[ \cdot \mu_4(m_p; n_q, l_1, l_2) \]
\[ - \mu_2(m_1; n_2) \mu_2(l_1, l_2) \]
\[ + \mu_2(k_1, k_2) \mu_2(m_1, m_2) \]
\[ \cdot \mu_4(l_1, l_2, n_1, n_2) \]
\[ - \mu_2(l_1, l_2) \mu_2(m_1, n_2) \]
\[ + \mu_2(k_1, k_2) \mu_2(n_1, n_2) \]
\[ - \mu_2(k_1, k_2) \mu_2(m_1, m_2) \]
\[ + \mu_2(k_1, k_2) \mu_2(n_1, n_2) \mu_4(k_1, k_2, m_1, m_2) \]
\[ - \mu_2(k_1, k_2) \mu_2(m_1, m_2) \] \[= 0. \]

This case, the expression in (81) simplifies to
\[ N_{a} \cdot E\{[\kappa_4(k_1, k_2; l_1, l_2) - \kappa_4(k_1, k_2; l_1, l_2)] \]
\[ \cdot \mu_6(k_1, k_2, m_1, m_2; l_1, l_2, n_1, n_2) \] \[ \approx \mu_6(k_1, k_2, m_1, m_2; l_1, l_2, n_1, n_2) \]
\[ - \sum_{p=1}^{2} \sum_{q=1}^{2} \mu_2(m_{3-p}; n_{3-q}) \]
\[ \cdot \mu_6(k_1, k_2, m_p; l_1, l_2, n_q) \]
\[ - \sum_{i=1}^{2} \sum_{j=1}^{2} \mu_2(k_{3-i}; l_{3-j}) \]
\[ \cdot \mu_6(m_1, m_2, k_i; n_1, n_2, l_j) \]
\[ + \sum_{i=1}^{2} \sum_{j=1}^{2} \sum_{p=1}^{2} \sum_{q=1}^{2} \mu_2(k_{3-i}; l_{3-j}) \]
\[ \cdot \mu_6(m_1, m_2, k_i; n_1, n_2, l_j) \]
\[ - \mu_2(m_3; n_3) \mu_4(k_1, m_p; l_j, n_q) \]
\[ - \mu_2(k_1, k_2) \mu_2(m_1, m_2) \mu_4(l_1, l_2) \]
\[ - \mu_2(k_1, k_2) \mu_2(m_1, m_2) \]
\[ - \mu_2(l_1, l_2) \mu_2(m_1, m_2) \]
\[ - \mu_4(m_1, m_2; n_1, n_2) \mu_2(l_1, l_2) \]
\[ + \mu_2(k_1, k_2) \mu_2(m_1, m_2) \]
\[ - \mu_2(k_1, k_2) \mu_2(m_1, m_2) \]
\[ \cdot \mu_4(l_1, l_2, n_1, n_2) \]
\[ - \mu_2(l_1, l_2) \mu_2(m_1, n_2) \]
\[ + \mu_2(k_1, k_2) \mu_2(n_1, n_2) \]
\[ - \mu_2(k_1, k_2) \mu_2(n_1, n_2) \mu_4(k_1, k_2, m_1, m_2) \]
\[ - \mu_2(k_1, k_2) \mu_2(m_1, m_2) \] \[= 0. \]

which is the equation derived in [10]. Given the true model of the measurements, all the moments can be computed. A method for computing the moments for statistically independent QAM signals in spatially white Gaussian noise is given in [10]. Alternatively, these moments can be estimated when only measurements are given.

**APPENDIX B**

**COVARIANCE OF EVEN-ORDER MOMENTS**

We assume that the received signals have zero first-order moments. We also assume that the signals emitted at different times by the same source or by different sources are statistically independent. The even-order moments are given by

\[ \mu_{2n}(k_1, \ldots, k_a; l_1, \ldots, l_b) \]
\[ = E \left\{ \prod_{i=1}^{a} z_{k_i}(t) \prod_{j=1}^{b} z_{l_j}(t) \right\} \] \[= E \left\{ \prod_{i=1}^{a} z_{k_i}(t) \prod_{j=1}^{b} z_{l_j}(t) \right\} \]

\[ \delta \mu_{2n}(k_1, \ldots, k_a; l_1, \ldots, l_b) \]
\[ = \mu_{2n}(k_1, \ldots, k_a; l_1, \ldots, l_b) \]
\[ - \mu_{2n}(k_1, \ldots, k_a; l_1, \ldots, l_b) \]

where the sum of the subscripts is \( a + b = 2n \) and \( n \) is an integer. The covariance of even order sample moments is, in general

\[ E\{\delta \mu_{2n}(k_1, \ldots, k_a; l_1, \ldots, l_b) \}

The indices \((k_1, k_2, l_1, l_2)\) of the first sample cumulant refer to the elements in the received data vector \( z(t) \). Likewise, the indices \((m_1, m_2, n_1, n_2)\) also refer to the elements of \( z(t) \).

For most QAM signals, the symbols are symmetrically distributed in such a way that the fourth-order cumulant is given by

\[ \kappa_4(k_1, k_2; l_1, l_2) = \mu_4(k_1, k_2; l_1, l_2) \]
\[ - \mu_2(k_1; l_1) \mu_2(k_2; l_2) \]
\[ - \mu_2(k_1; l_2) \mu_2(k_2; l_1) \] \[= 0. \]

That is, the unconjugated second-order moments are zero.

\[ \mu_2(k_1, k_2) = E\{z_{k_1} z_{k_2} \} \]
\[ = 0. \]
\[ E \{ \delta \mu_2(h; l) \cdot \delta \mu_2(m; n) \} = \frac{1}{N_a} \mu_2(k; l, n) - \frac{1}{N_a} \mu_2(k; l) \cdot \mu_2(m; n). \] (88)

Recall that we have assumed zero-mean processes in this paper, so that the second order moment \( \mu_2(\cdot) \) is equal to its corresponding covariance.

REFERENCES


Norman Yuen grew up in Oakland, CA. He received the B.S. and M.S. degrees in electrical engineering from the University of California, Berkeley, CA, in 1982 and 1986, respectively. He received the Ph.D. degree in electrical engineering from the University of California, Davis (UCD), in 1995.

In between academic degrees, he has worked as an engineer at National Semiconductor, Electronic Support Systems, and Sun Microsystems. As a research assistant at UCD, he actively studied array processing methods using higher order statistics.

He is currently working as a postdoctoral researcher at UCD. His research interests include sensor array processing, application of higher order statistics, blind signal separation, channel equalization, and performance analysis of array processing methods.

Dr. Yuen is a member of Tau Beta Pi and Eta Kappa Nu.

Benjamin Friedlander (S’74–M’76–SM’82–F’87) received the B.Sc. and M.Sc. degrees in electrical engineering from Technion–Israel Institute of Technology in 1968 and 1973, respectively. He received the Ph.D. degree in electrical engineering and the M.Sc. degree in statistics from Stanford University in 1976.

From 1976 to 1985, he was with Systems Control Technology, Inc., Palo Alto, CA. From November 1985 to July 1988, he was with Sassy Computer Corporation, Sunnyvale, CA. Currently, he is with the University of California, Davis, CA, and Signal Processing Technology, Ltd., Palo Alto.

Dr. Friedlander is the recipient of the 1983 ASSP Senior Award, the 1985 Award for the Best Paper of the Year from the European Association for Signal Processing (EURASIP), and the 1989 Technical Achievement Award of the Signal Processing Society.