Sinusoidal Frequency Estimation by Signal Subspace Approximation

Juha T. Karhunen, *Member, IEEE*, and Jyrki Joutsensalo

***Abstract*—Eigenvector-based methods such as multiple signal classification (MUSIC) are currently popular in sinusoidal frequency estimation due to their high resolution. A problem with these methods is the often high cost of estimating the eigenvectors of the autocorrelation matrix spanning the signal (or noise) subspace. In this work, we propose an efficient Fourier transform-based method avoiding eigenvector computation for approximating the signal subspace. The resulting signal subspace estimate can be used directly to define a MUSIC-type frequency estimator or as a very good initial guess in context with adaptive or iterative eigenvector computation schemes. At low signal-to-noise ratios, the approximation yields better results than exact MUSIC. It is also more robust than MUSIC against overestimating the number of sinusoids. Some variations of the basic method are briefly discussed.**

I. Introduction

In several applications, the problem of estimating the frequencies of multiple sinusoids in additive white noise arises naturally. For this purpose, several spectral estimation techniques [13], [15] based on Fourier transforms or autoregressive modeling can be used. During the last few years, however, eigenvector-based methods [8], [13], [15], such as MUSIC, have received a lot of attention in frequency estimation and sensor array processing. These methods are based on a specific model of sinusoids in additive white noise and often provide superior resolution when the number of samples is small or the frequencies to be estimated are close to each other. On the other hand, they are much less complicated than the optimal nonlinear maximum likelihood method.

In this paper, we concentrate on the multiple signal classification (MUSIC) method proposed by Schmidt [20], [21]. MUSIC is currently probably the most popular of the eigenvector-based frequency estimators. Recent theoretical studies [18], [22] show that in many cases it yields asymptotically nearly optimal results, which provides justification for its generally good performance.

A problem with eigenvector-based frequency estimators is the often high cost associated with computing the necessary eigenvectors. This is particularly notable if the dimensionality of the eigenvectors is large or if one tries to track slow changes in the frequencies. Since most eigenvector-based frequency estimators are defined in terms of either principal (spanning the signal subspace) or nonprincipal eigenvectors (i.e., those corresponding to the smallest eigenvalues) of the autocorrelation matrix, only a possibly small subset of eigenvectors is needed in the actual computations.

Other problems with MUSIC are that its performance is considerably degraded at low signal-to-noise ratios and that the number of sinusoids should be known in advance [13]. In practice, such *a priori* knowledge is seldom available. If the signal-to-noise ratio is high, relative magnitudes of the eigenvalues of the autocorrelation matrix often give a good indication of the number of sinusoidal components, but this simple criterion becomes very unreliable in the case of strong noise. Better tests have been proposed (e.g., [5]), but these generally require a lot of additional computation.

A motivation of this study was the authors' wonder at the discrepancy in performance between eigenvector methods and Fourier transform methods in the areas of spectral estimation and image processing. In image coding [1], the optimal (in the mean-square error sense) eigenvector-based Karhunen–Loève transform is used mainly as a reference only because it performs slightly better than such fast transforms as the discrete cosine transform (DCT) or discrete Fourier transform (DFT) for most practical images (e.g., [16]). On the other hand, in sinusoidal frequency estimation, eigenvector-based methods generally provide clearly higher resolution and better accuracy than classical Fourier-based spectral estimators.

In this paper, we propose methods for approximating the signal subspace in terms of a generalization of the DFT (or DCT) transform. The resulting signal subspace estimate yields very similar results as the standard one defined by the estimated principal eigenvectors in context with MUSIC. This shows that the high resolution of MUSIC (and other eigenvector-based methods) stems mainly from the form of the estimator. By using Fourier transform like eigenvectors, one can achieve high resolution. This is the main result of this paper. Furthermore, the proposed version of MUSIC seems to be more robust than the standard one and has certain computational advantages.

The problem of approximating the signal subspace without eigendecomposition has not been dealt much in the literature. Kay and Shaw [14] propose a functional approximation method quite different from ours. At the end of this paper, the possibility of designing "crude"

Manuscript received March 8, 1990; revised June 27, 1991.
The authors are with the Helsinki University of Technology, Laboratory of Computer and Information Sciences, SF-02150 Espoo, Finland.
IEEE Log Number 9203234.

1053-587X/93$3.00 © 1992 IEEE

eigenfilters is conjectured but not studied. The work by Tufts and Melissinos [23] seems to be most closely related to ours. They use prior information from nonparametric spectrum analysis or beamforming to generate a starting vector, from which the signal eigenvectors are estimated using the Prony–Lanczos algorithm. This procedure improves the signal subspace estimates at low SNR's, but is less general than our approach.

This paper is arranged as follows. In Section II, the necessary background for eigenanalysis-based frequency estimation is shortly presented. In the next section, the new signal subspace approximation is introduced and justified. Section IV consists of experimental results, followed by a more general discussion in the next section. The last section contains the main conclusions and some remarks.

II. Eigenanalysis-Based Frequency Estimation

A. MUSIC Frequency Estimator

The basic theory of eigenvector-based frequency estimation and the MUSIC method is well represented in, e.g., [8], [13], [15]. In the following, the results needed later in this paper are reviewed.

The data are assumed to consist of M complex sinusoids in additive white noise. The N available data samples $x[0], \cdots, x[N-1]$ are thus modeled as

$$x[k] = \sum_{m=1}^{M} A_m e^{j(2\pi f_m k + \theta_m)} + w[k] \tag{1}$$

where the amplitudes A_m and frequencies f_m of the sinusoids are unknown constants. The phases θ_m are assumed to be uniformly distributed on the interval $[0, 2\pi)$. The complex white noise term $w[k]$ has zero mean and variance σ^2.

In practical estimation data vectors

$$\boldsymbol{x}_k = (x[k], x[k+1], \cdots, x[k+L-1])^T \tag{2}$$

formed of L successive samples are used. Their $L \times L$ autocorrelation matrix can be shown to be

$$\boldsymbol{R}_{xx} = E(\boldsymbol{x}_k \boldsymbol{x}_k^H) = \sum_{m=1}^{M} A_m^2 \boldsymbol{e}_m \boldsymbol{e}_m^H + \sigma^2 \boldsymbol{I}. \tag{3}$$

Here the signal (or frequency information) vectors $\boldsymbol{e}_m$, corresponding to the frequencies of the sinusoids, are defined as

$$\boldsymbol{e}_m = (1, e^{j2\pi f_m}, \cdots, e^{j2\pi f_m(L-1)})^T. \tag{4}$$

It would be desirable to extract the signal vectors from (3), but this is not directly possible, even though the theoretical correlation matrix $\boldsymbol{R}_{xx}$ is known exactly. However, it can be shown that the M principal eigenvectors $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_M$ of $\boldsymbol{R}_{xx}$ span the same signal subspace as the signal vectors $\boldsymbol{e}_1, \cdots, \boldsymbol{e}_M$. The remaining eigenvectors $\boldsymbol{u}_{M+1}, \cdots, \boldsymbol{u}_L$ correspond to noise, and span the so-called noise subspace. This decomposition is essential in defining various eigenvector-based frequency estimators.

Consider now in particular the MUSIC frequency estimator. In practice, it is realized typically as follows:

1) Estimate $\boldsymbol{R}_{xx}$ from the data vectors $\boldsymbol{x}_0, \cdots, \boldsymbol{x}_{K-1}$. Some estimation methods are given in the next subsection.

2) Compute the eigenvectors $\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2, \cdots, \hat{\boldsymbol{u}}_L$ of the estimated correlation matrix $\hat{\boldsymbol{R}}_{xx}$. Choose the M principal eigenvectors $\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2, \cdots, \hat{\boldsymbol{u}}_M$ (corresponding to the M largest eigenvalues) as the basis of the signal subspace.

3) The sinusoidal frequencies are obtained as peak locations of the MUSIC frequency estimator

$$\hat{P}(f) = \frac{1}{L - \sum_{i=1}^{M} |\boldsymbol{e}_f^H \hat{\boldsymbol{u}}_i|^2}. \tag{5}$$

Here, $\boldsymbol{e}_f$ is the signal vector corresponding to the frequency f.

Alternatively, the MUSIC frequency estimator is often defined in terms of the (estimated) nonprincipal eigenvectors $\hat{\boldsymbol{u}}_i$, $i = M+1, \cdots, L$:

$$\hat{P}(f) = \frac{1}{\sum_{i=M+1}^{L} |\boldsymbol{e}_f^H \hat{\boldsymbol{u}}_i|^2}. \tag{6}$$

Theoretically, both computational forms of MUSIC have infinite values at the exact frequencies of the sinusoids. In practice, sharp peaks near the correct frequencies are the best we can have. The denominator in (5) or (6) can be evaluated effectively at a large number of frequencies in terms of the FFT algorithm.

Knowing the correct number M of sinusoids is important in MUSIC. If M is underestimated, some sinusoids are missed and the estimated frequencies tend to be incorrect. If M is too large in (5) or (6), spurious frequency peaks appear in addition to correct ones.

B. Eigenvector Estimation

In practice, the theoretical correlation matrix $\boldsymbol{R}_{xx}$ or its eigenvectors $\boldsymbol{u}_i$ are not known, but must be estimated from the available samples. Usually this is done via an off-line procedure, in which $\boldsymbol{R}_{xx}$ is first estimated in terms of the data vectors $\boldsymbol{x}_0, \cdots, \boldsymbol{x}_{K-1}$ as follows:

$$\hat{\boldsymbol{R}}_{xx} = \frac{1}{K} \sum_{k=0}^{K-1} \boldsymbol{x}_k \boldsymbol{x}_k^H. \tag{7}$$

After this, the eigenvectors $\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2, \cdots, \hat{\boldsymbol{u}}_L$ of $\hat{\boldsymbol{R}}_{xx}$ are computed using some standard method of numerical analysis. Here $K = N - L + 1$ is the number of the data vectors $\boldsymbol{x}_k$ that can be formed from the available N scalar samples $x[0], \cdots, x[N-1]$ using (2).

Instead of the simple backward-only estimate (7), the forward–backward estimate

$$\hat{\boldsymbol{R}}_{xx} = \frac{1}{2K} \sum_{k=0}^{K-1} (\boldsymbol{x}_k \boldsymbol{x}_k^H + \tilde{\boldsymbol{x}}_k \tilde{\boldsymbol{x}}_k^H) \tag{8}$$

$$\tilde{\boldsymbol{x}}_k = (x^*[k+L-1], \cdots, x^*[k+1], x^*[k])^T \tag{9}$$

is often used for vectors having the special structure (2), since it yields better resolution. Different ways of estimating the correlation matrix and other refinements lead to a number of variants of MUSIC (e.g., [3], [22]). Since several of these variants are rather involved, and our goal is to develop a computationally efficient approximation to MUSIC, we concentrate on the basic estimator (7).

It is often desirable that either the signal or noise eigenvectors could be updated using incoming new samples. To this end, several data-adaptive eigenvector estimation algorithms of different complexity and accuracy have been proposed. Relevant references and comparisons of various algorithms can be found in [2], [3], [9], [10], [25].

Theoretically, it doesn't matter whether a frequency estimator is defined in terms of signal or noise subspace. If the eigenvectors of the sample correlation matrix are computed with high accuracy, the results are identical in practice, too. Using some lower accuracy gradient-type algorithm for adaptive eigenvector estimation often produces, however, substantially different results. In general, estimation of principal eigenvectors is easier, and they are more robust against noise and other disturbances [8], [9]. From (6) it is seen that the number of eigenvectors $L - M$ needed grows linearly with the length L of the data vectors, if the MUSIC estimator is computed using the noise subspace formulation. But for the signal subspace-based estimator [5], the number of eigenvectors M remains fixed. This allows the use of longer data vectors $\boldsymbol{x}_k$ that have better frequency resolution properties [19], [22] without increasing the computational load too much. For these reasons, the signal subspace formulation (5) is used in this paper.

III. Signal Subspace Approximation

A. The Approximation Method

Our method of approximating the signal subspace is based on a generalization of such well-known fast transforms as the DFT or DCT [1], [4], [7] to vectorial data. For the DFT, the generalization is essentially the same as the multichannel Fourier transform (e.g., [13, chapter 14]) applied to data vectors (2). The approximation method is as follows.

1) From the available scalar samples, construct J successive data vectors $\boldsymbol{x}_0, \boldsymbol{x}_1, \cdots, \boldsymbol{x}_{J-1}$ according to (2). Here J is often chosen as a power of two, and/or $J = K$.

2) Transform the data vectors:

$$\boldsymbol{v}_i = \sum_{k=0}^{J-1} g_{ik}\boldsymbol{x}_k, \qquad i = 0, 1, \cdots, J-1. \tag{10}$$

Here g_{ik} is the chosen forward transform kernel; for the DFT, it is defined as $g_{ik} = e^{-j2\pi ik/J}$; for the DCT, $g_{ik} = \cos(\pi ik/J)$. The vectors $\boldsymbol{v}_i$ can be evaluated effectively using fast transforms; see Section V-B.

3) Compute the squared norms $\|\boldsymbol{v}_i\|^2$, $i = 0, 1, \cdots, J-1$. Choose the M $\boldsymbol{v}_i$-vectors having the largest norms as the basis that defines the approximation to M-dimensional signal subspace. These basis vectors are denoted by $\boldsymbol{v}_1', \boldsymbol{v}_2', \cdots, \boldsymbol{v}_M'$.

4) To simplify further processing, it is often necessary to orthonormalize the chosen basis. The resulting orthonormal basis is denoted by $\boldsymbol{o}_1, \boldsymbol{o}_2, \cdots, \boldsymbol{o}_M$. Since this defines the same signal subspace approximation as the $\boldsymbol{v}_i'$-vectors, the method of orthonormalization is not important; e.g., the well-known Gram–Schmidt procedure may be used.

The proposed approximation may be used directly for computing the MUSIC-type frequency estimator

$$\hat{P}(f) = \frac{1}{L - \sum_{i=1}^{M} |\boldsymbol{e}_f^H \boldsymbol{o}_i|^2}. \tag{11}$$

This is the same form as (5), but the estimated principal eigenvectors have been replaced by the orthonormalized basis vectors of the signal subspace approximation.

Even though it turns out that (11) is more robust than standard MUSIC against overestimation of the number M of the sinusoids, some kind of estimate of M is often needed. A simple estimate may be obtained by inspecting the relative magnitudes of the squared norms in step 3) above. In our approximation method, the squared norms have the same role as eigenvalues in MUSIC. Thus, the squared norms corresponding mainly to the noise are roughly of the same size and smaller than those corresponding to the sinusoids. A conservative estimate of M is the number of squared norms significantly larger than the smaller ones. In practice, it is better to choose at least a slightly larger value of M in order to avoid missing any sinusoids. See the experimental results in Section IV.

In the following subsections, we present theoretical considerations that justify the approximation method. Before proceeding, a few words about the assumed model (1). In our method, the phases θ_m are fixed constants rather than uniformly distributed random numbers. The white noise assumption is not necessary, but the approximation method obviously works better in this case. Otherwise, the model is the same as in MUSIC.

B. Connection with the Principal Eigenvectors

The M-dimensional signal subspace is defined as the space spanned by the M principal eigenvectors $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_M$, corresponding to the M largest eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_M$ of the theoretical autocorrelation matrix $\boldsymbol{R}_{xx}$. The defining equation is

$$\boldsymbol{R}_{xx}\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i, \qquad \|\boldsymbol{u}_i\| = 1. \tag{12}$$

Inserting (7) into (12) yields

$$\lambda_i \boldsymbol{u}_i \approx \frac{1}{K}\sum_{k=0}^{K-1} (\boldsymbol{x}_k^H \boldsymbol{u}_i)\boldsymbol{x}_k. \tag{13}$$

Thus, the true eigenvectors are approximately some linear combinations of the data vectors $\boldsymbol{x}_k$. For the eigenvectors

$\hat{\boldsymbol{u}}_i$ of the sample correlation matrix (7), the formula (13) holds exactly. The first principal eigenvector $\hat{\boldsymbol{u}}_1$ is the unit vector producing the linear combination (13) of the data vectors having the maximum norm (eigenvalue) $\hat{\lambda}_1$. The second eigenvector $\hat{\boldsymbol{u}}_2$ must be orthogonal to the first one, and defines the linear combination with the maximum norm $\hat{\lambda}_2$ among such orthogonal directions, and so on.

The connection with our approximation method is clear: in (10), we also form linear combinations of the data vectors and choose those linear combinations having the largest norms. The weights in (10) are obtained from vectors

$$\boldsymbol{g}_i = (g_{i0}, g_{i1}, \cdots, g_{i(J-1)})^T \tag{14}$$

that are usually chosen to be mutually orthogonal: $\boldsymbol{g}_i^H \boldsymbol{g}_j = 0$, $i \neq j$. It is not essential that the vectors $\boldsymbol{g}_i$ are normalized to unity, but they should have equal norms.

To find good approximations to the principal eigenvectors, one should, in general, make a complete search over the L-dimensional hypersphere defined by $\|\boldsymbol{g}\| = $ constant in (10). This is computationally prohibitive, even though iterative refinement could be applied. The proposed approximation method is based on the fact that the vectors $\boldsymbol{x}_k$ contain information about the frequencies of the sinusoids according to the assumed model (1). The vectors $\boldsymbol{g}_0, \cdots, \boldsymbol{g}_{J-1}$ are chosen in such a way that each of them corresponds to some frequency in the normalized frequency interval. The natural choice is to divide the normalized frequency interval $[-0.5, 0.5)$ in J evenly spaced discrete values, and let the vector $\boldsymbol{g}_{i-1}$ represent the ith discrete frequency.

C. Change in the Signal-to-Noise Ratio

Using definitions (1), (2), and (4), the data vectors can be written in the form

$$\boldsymbol{x}_k = \sum_{m=1}^{M} A_m e^{j(2\pi f_m k + \theta_m)} \boldsymbol{e}_m + \boldsymbol{w}_k \tag{15}$$

where

$$\boldsymbol{w}_k = (w[k], w[k+1], \cdots, w[k+L-1])^T \tag{16}$$

is the noise vector. The signal-to-noise ratio of the original data samples, or the components of the vectors (15), is defined as

$$\text{SNR}(\boldsymbol{x}) = 10 \log \frac{\sum_{m=1}^{M} A_m^2}{2\sigma^2}. \tag{17}$$

Inserting (15) into (10), we obtain

$$\boldsymbol{v}_i = \sum_{k=0}^{J-1} \sum_{m=1}^{M} g_{ik} A_m e^{j(2\pi f_m k + \theta_m)} \boldsymbol{e}_m + \sum_{k=0}^{J-1} g_{ik} \boldsymbol{w}_k. \tag{18}$$

What is the corresponding signal-to-noise ratio for the components of the transformed vectors (18)? Assuming white noise here, the components of the transformed noise terms $\sum_{k=0}^{J-1} g_{ik} \boldsymbol{w}_k$ have zero mean and their variance is

$$\sigma_i^2 = \sum_{k=0}^{J-1} |g_{ik}|^2 \sigma^2 = \|\boldsymbol{g}_i\|^2 \sigma^2. \tag{19}$$

The squares $A_m^2 = |A_m e^{j(2\pi f_m k + \theta_m)}|^2$ of the weights of the frequency information vectors $\boldsymbol{e}_m$ in (15) must be replaced by

$$B_m^2 = \left| \sum_{k=0}^{J-1} g_{ik} e^{j2\pi f_m k} \right|^2 A_m^2. \tag{20}$$

Thus, the new signal-to-noise ratio is

$$\text{SNR}(\boldsymbol{v}_i) = 10 \log \frac{\sum_{m=1}^{M} B_m^2}{2\sigma_i^2}. \tag{21}$$

In the special case of equal-amplitude sinusoids: $A_1 = A_2 = \cdots = A_M = A$, (21) can be simplified further

$$\begin{aligned} \text{SNR}(\boldsymbol{v}_i) &= 10 \log \left(\frac{\sum_{m=1}^{M} \left| \sum_{k=0}^{J-1} g_{ik} e^{j2\pi f_m k} \right|^2}{M \|\boldsymbol{g}_i\|^2} \cdot \frac{MA^2}{2\sigma^2} \right) \\ &= 10 \log \left(\frac{\sum_{m=1}^{M} \left| \sum_{k=0}^{J-1} g_{ik} e^{j2\pi f_m k} \right|^2}{M \|\boldsymbol{g}_i\|^2} \right) \\ &\quad + \text{SNR}(\boldsymbol{x}). \end{aligned} \tag{22}$$

From (22), one can compute the effect of the transformation to the signal-to-noise ratio at various frequencies. If we consider only one sinusoid ($M = 1$), the increase/decrease in the SNR is

$$S_i(f_m) = 10 \log \left(\frac{\left| \sum_{k=0}^{J-1} g_{ik} e^{j2\pi f_m k} \right|^2}{\|\boldsymbol{g}_i\|^2} \right). \tag{23}$$

If the vectors (14) are chosen so that each of them represents some frequency (as is the case when the DFT or DCT basis vectors are used), the transformation (10) increases the signal-to-noise ratio at that frequency and near it, and decreases the SNR at other frequencies. For the DFT kernel $g_{ik} = e^{-j2\pi ik/J}$, the argument of the logarithm in (23) simplifies to $\sin^2(\omega J)/[J \sin^2(\omega)]$, $\omega = \pi(i/J - f_m)$, and the maximum value $10 \log J$ of (23) is attained at $f_m = i/J$.

It is now evident that if we are able to select from the transformed vectors (10) those that reinforce the correct frequencies of the sinusoids, an approximation to the signal subspace is obtained.

The squared norms $\|\boldsymbol{v}_i\|^2$ of all the transformed vectors $\boldsymbol{v}_i$, $i = 0, \cdots, J-1$, could be used directly as a spectral estimator, but the resolution of this estimator is low. In fact, one can rather easily show that for the DFT kernel, the squared norms define a Bartlett-type estimator: $\|\boldsymbol{v}_i\|^2 = L\boldsymbol{e}_i^T \hat{\boldsymbol{R}}_{yy} \boldsymbol{e}_i^*$, where $\boldsymbol{e}_i$ is of the form (4) with L replaced by J and $\hat{\boldsymbol{R}}_{yy}$ is the estimated correlation matrix of the vectors $y_k = (x^*[k], \cdots, x^*[k+J-1])^T$, $k = 1, \cdots, L$.

D. Least Squares Formulation

Consider now more closely the selected vectors $v'_1, \cdots, v'_M$ that define the signal subspace approximation. For them, (18) may be rewritten in a clearer form

$$v'_i = \sum_{m=1}^{M} \alpha'_{im} e_m + n'_i, \qquad i = 1, \cdots, M. \quad (24)$$

With the definitions $\alpha_{im} = \Sigma_{k=0}^{J-1} g_{ik} A_m e^{j(2\pi f_m k + \theta_m)}$ and $n_i = \Sigma_{k=0}^{J-1} g_{ik} w_k$, this of course holds for all the v_i-vectors. However, the error vectors n'_i are relatively smaller for the selected vectors. Assuming linear independence, one can solve the signal vectors from (24) and express them, in turn, as linear combinations of the v'_i vectors and an error vector z_m

$$e_m = \sum_{i=1}^{M} \beta_{mi} v'_i + z_m, \qquad m = 1, \cdots, M. \quad (25)$$

A natural way of determining the unknown coefficients β_{mi}, $i = 1, \cdots, M$, of the expansion (25) for each e_m is to use the well-known least squares estimation method. The minimum least squares error corresponding to the optimal coefficients can be shown to be (e.g., [13, pp. 48–49])

$$\|z_m\|^2 = e_m^H[I - V(V^H V)^{-1} V^H] e_m$$
$$= L - e_m^H V(V^H V)^{-1} V^H e_m. \quad (26)$$

Here $V = [v'_1, \cdots, v'_M]$ is a matrix collected from the v'_i vectors.

Orthonormalization of the vectors $v'_1, \cdots, v'_M$ does not affect the least squares error (26) in any way. It is easy to see this by expressing the orthonormalized basis $O = [o_1, \cdots, o_M]$ in the form $O = VS$, where S is the nonsingular square matrix defining the transformation. However, orthonormalization simplifies greatly the computations since $O^H O = I$, and (26) becomes

$$\|z_m\|^2 = L - \sum_{i=1}^{M} |e_m^H o_i|^2. \quad (27)$$

But just this function appears in the denominator of (11). Since the signal vectors e_m are actually unknown, the least squares fitting must be done for each possible frequency (signal vector). The best matching frequencies appear then as the peak locations of the frequency estimator (11).

From the discussion above, some important conclusions can be made:

1) The proposed MUSIC-type estimator is optimal in the least squares sense for the model (24).

2) Any basis of the same subspace yields the same MUSIC estimator [defined as the inverse of the error criterion (27) or (26)].

3) It is not necessary that each selected basis vector v'_i is a good approximation to some single signal vector or principal eigenvector. It suffices that the set $v'_1, \cdots, v'_M$ of basis vectors approximates well the signal subspace.

The high-resolution properties of our method follow essentially from conclusions 1) and 3). Classical Fourier-based spectral estimators such as the periodogram are able to resolve the sinusoids provided that their frequencies are not too close to each other [13]. Roughly speaking, this corresponds to the situation where in (24) only one of the coefficients α_{im} is large for each i and the others are considerably smaller or negligible. Least squares fitting to the estimated signal subspace allows the vectors v'_i to be linear combinations of several signal vectors, and yet it is often possible to resolve the corresponding sinusoids.

IV. Experimental Results

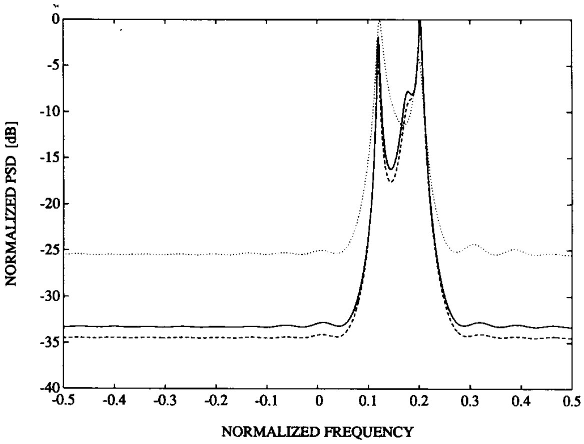
A number of simulations have been made in order to test the ideas and compare the approximation to the MUSIC frequency estimator. In these simulations, the samples $x[k]$ and data vectors x_k were generated from (1) and (2), respectively. The white noise term $w[k]$ was Gaussian. The MUSIC frequency estimators were computed from (5) using either the backward estimate (7) or the forward–backward estimate (8), and the corresponding approximation was obtained from (11). We used the DFT for computing the basis vector candidates in (10); the resulting frequency estimator is abbreviated DFT-MUSIC. The resulting pseudospectra were normalized so that their peak value was unity or 0 dB.

In the first test case, the data was complex and consisted of three sinusoids at normalized frequencies $f_1 = 0.12$, $f_2 = 0.18$, and $f_3 = 0.20$ in white noise. The amplitudes of the sinusoids were, respectively, $A_1 = 1$, $A_2 = 1$, $A_3 = 2$, and their phases were zero. The dimensionality of the data vectors x_k was $L = 15$. Figs. 1–3 show experimental results with this data. For an idea of the variance, frequency estimators given by three different data sequence realizations are depicted in each subpicture.

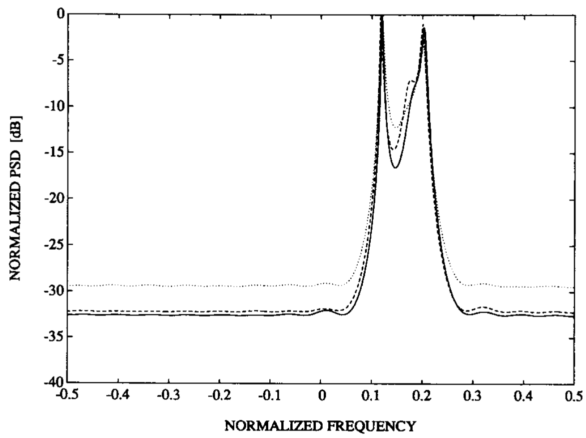
In Fig. 1, the number of samples was small, $N = 30$ and the signal-to-noise ratio was good, 20 dB. The signal subspace dimensionality had the correct value $M = 3$. In this case, the forward–backward MUSIC yields the best resolution [Fig. 1(c)], whereas the backward-MUSIC [Fig. 1(b)] and the DFT-MUSIC [Fig. 1(a)] produce almost equal curves.

Fig. 2(a)–(c) shows the corresponding results when the number of samples was increased to $N = 46$. From Fig. 2(e), it is seen that a slight overestimation ($M = 4$) of the number of the sinusoids has improved the resolution of the DFT-MUSIC clearly while keeping off the false peaks. The result is at least as good as that of backward-MUSIC with the correct dimension $M = 3$ [Fig. 2(b)] or overestimated dimension $M = 4$ [Fig. 2(d)].

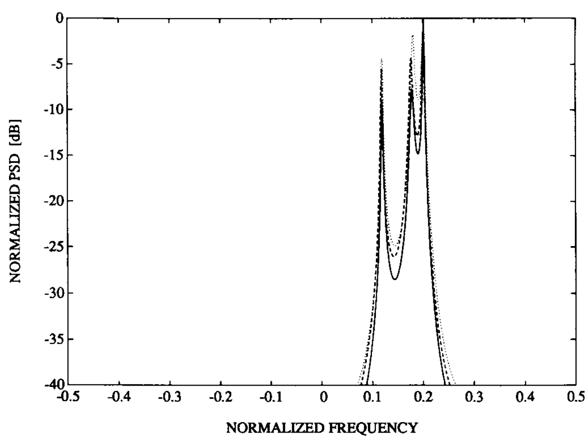
In Fig. 3, the signal-to-noise ratio was low (0 dB), and the number of samples was relatively large ($N = 142$). In this case, the DFT-MUSIC method yields the best frequency estimators. With the correct dimensionality $M =$



(a)

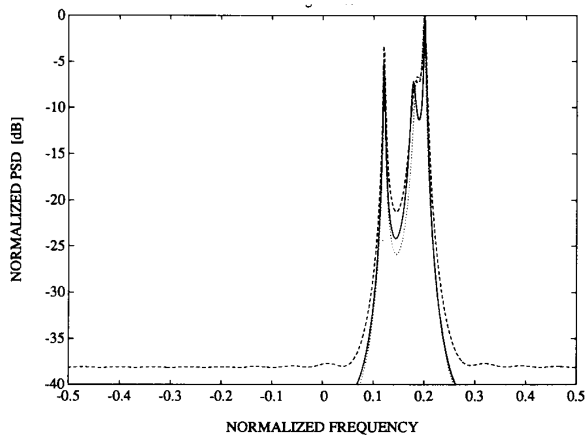


(b)

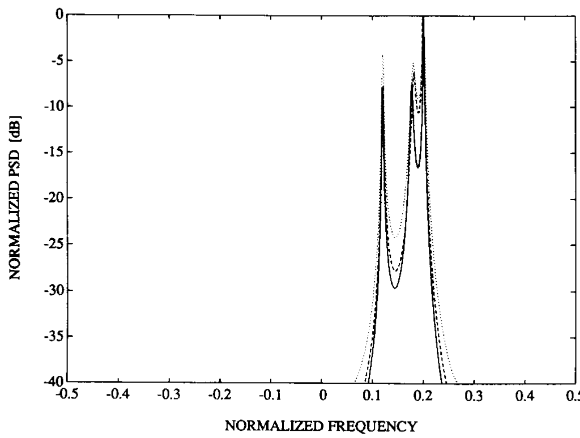


(c)

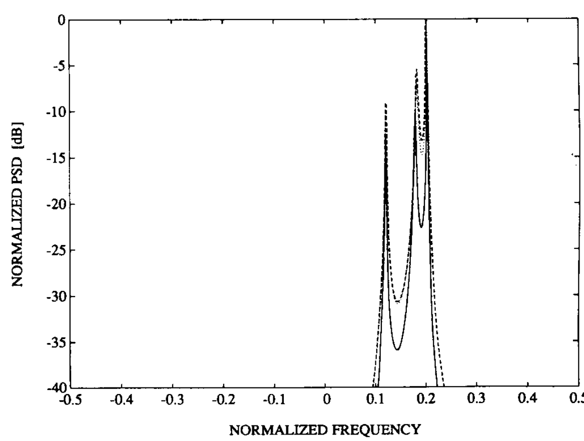
Fig. 1. Various frequency estimators. Data: three complex sinusoids at normalized frequencies 0.12, 0.18, and 0.20 in white noise. SNR = 20 dB, three independent sets of 30 samples. Dimensionality of data vectors $L = 15$: (a) DFT-MUSIC, correct signal subspace dimensionality $M = 3$; (b) MUSIC, backward estimator, correct dimensionality $M = 3$; and (c) MUSIC, forward–backward estimator, correct dimensionality $M = 3$.



(a)



(b)



(c)

Fig. 2. Various frequency estimators. Data: three complex sinusoids at normalized frequencies 0.12, 0.18, and 0.20 in white noise. SNR = 20 dB, three independent sets of 46 samples. Dimensionality of data vectors $L = 15$: (a) DFT-MUSIC, correct signal subspace dimensionality $M = 3$; (b) MUSIC, backward estimator, correct dimensionality $M = 3$; (c) MUSIC, forward–backward estimator, correct dimensionality $M = 3$. (*Continued on next page.*)

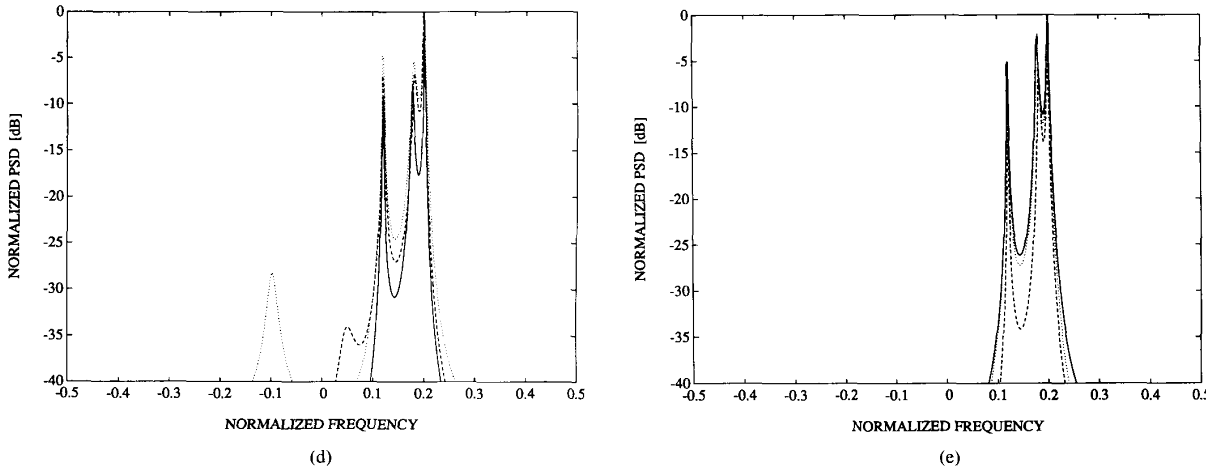


Fig. 2. (*Continued.*) (d) MUSIC, backward estimator, overestimated dimensionality $M = 4$. (e) DFT-MUSIC, overestimated dimensionality $M = 4$.

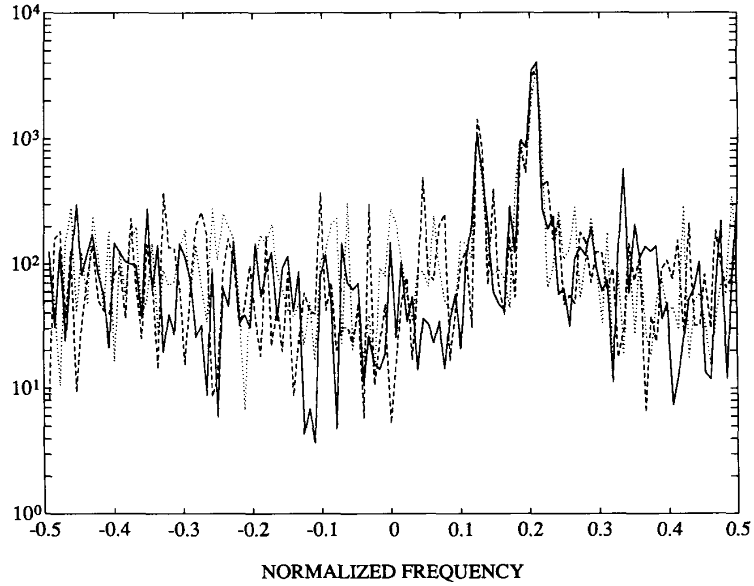


Fig. 3. Various frequency estimators. Data: three complex sinusoids at normalized frequencies 0.12, 0.18, and 0.20 in white noise. SNR = 0 dB, three independent sets of 142 samples. Dimensionality of data vectors $L = 15$: (a) Squared norms of DFT transformed data vectors. (*Continued on next page.*)

3, it cannot yet resolve the frequencies $f_2 = 0.18$ and $f_3 = 0.20$ [Fig. 3(b)] but does not show any false frequencies as is the case with the MUSIC estimators [Fig. 3(c) and (d)]. Again, a slight overestimation of the signal subspace dimensionality leads to an excellent result [Fig. 3(f)]. The reason of this improvement is that a fourth $\boldsymbol{v}_i$-vector corresponding roughly to the frequency $f = 0.18$ has been included into the computation of DFT-MUSIC. This is seen by inspecting the squared norms of the tentative basis vectors in Fig. 3(a). When the dimensionality of the signal subspace is increased to $M = 8$, the MUSIC estimator becomes very unreliable but anyway cannot resolve the close frequencies [Fig. 3(g)], while the DFT-MUSIC estimator is still rather reliable and useful [Fig. 3(h)].

In these experiments, the dimensionality of the data vectors was held in a fixed moderate value $L = 15$ without trying to optimize it. Especially in the strong noise case, resolution is possible using less samples if L is taken larger. However, this increases the computational load rapidly; see Section V-B.

A numerical performance comparison is given in Tables I–IV. For these simulations, 100 independent realizations of 40 samples of two complex sinusoids in white

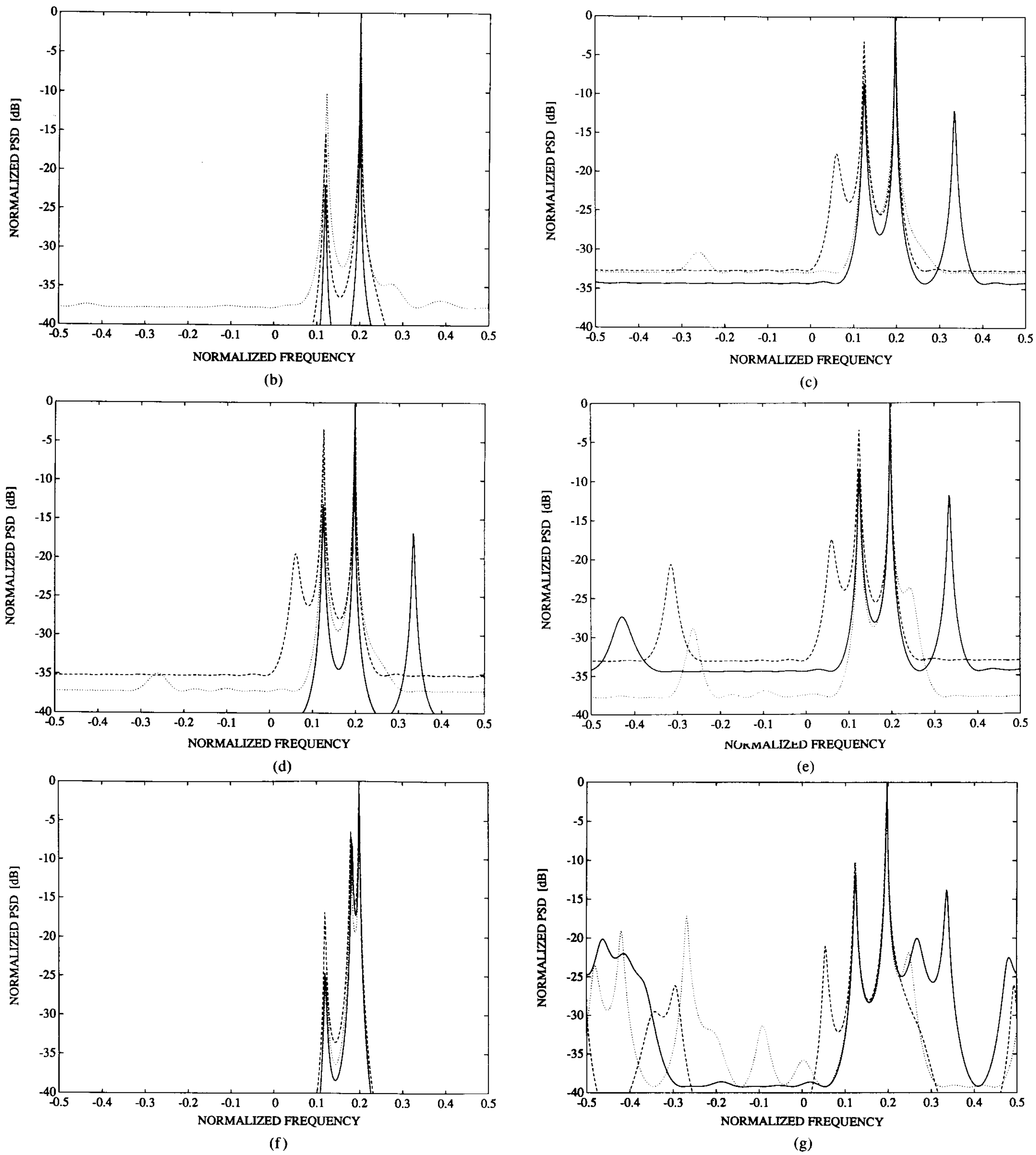


Fig. 3. (*Continued.*) (b) DFT-MUSIC, correct signal subspace dimensionality $M = 3$. (c) MUSIC, backward estimator, correct dimensionality $M = 3$. (d) MUSIC, forward-backward estimator, correct dimensionality $M = 3$. (e) MUSIC, backward estimator, overestimated dimensionality $M = 4$. (f) DFT-MUSIC, overestimated dimensionality $M = 4$. (g) MUSIC, backward estimator, overestimated dimensionality $M = 8$. (*Continued on next page.*)

noise were generated. The normalized frequencies of the sinusoids were chosen otherwise randomly but so that the constraint $|f_1 - f_2| = 0.02$ was satisfied. Thus, the two sinusoids were spaced closer than $1/N = 0.025$, the Fourier resolution. The phases of the sinusoids were chosen randomly and their amplitudes were random numbers in the interval $(1, 4)$. The signal subspace had the correct dimensionality $M = 2$ in Tables I–III, and the length $L = 17$ of the data vectors was chosen so that it was roughly optimal for all the methods tested. The frequency spacing

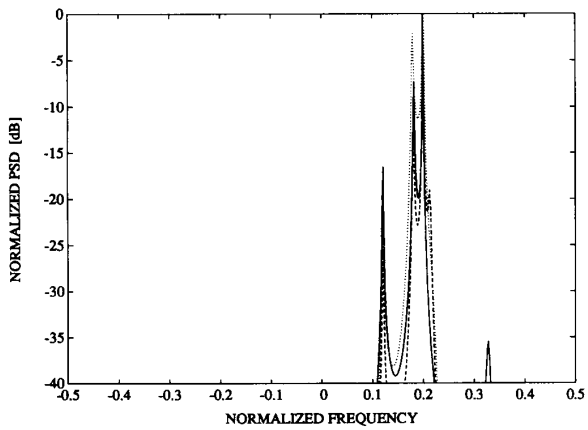


Fig. 3. (*Continued.*) (h) DFT-MUSIC, overestimated dimensionality $M = 8$.

TABLE I
NUMBER OF SUCCESSFUL EXPERIMENTS (RESOLVED CASES) AT VARIOUS SIGNAL-TO-NOISE RATIOS OUT OF 100 EXPERIMENTS WITH THE VARIOUS METHODS. DATA: 40 SAMPLES OF TWO COMPLEX SINUSOIDS IN WHITE NOISE, NORMALIZED FREQUENCIES SEPARATED BY 0.02. CORRECT SIGNAL SUBSPACE DIMENSIONALITY $M = 2$

Signal-to-Noise Ratio (dB)	−4	0	4	8	12	16	20
MUSIC (forward–backward)	4	13	37	67	89	100	100
MUSIC (backward)	5	8	16	27	69	92	100
DFT-MUSIC (norm)	11	25	30	46	65	80	95
DFT-MUSIC (periodogram)	13	18	25	38	69	92	98
ACM-MUSIC (norm)	24	46	61	77	90	98	100
ACM-MUSIC (periodogram)	18	31	60	83	90	99	100

TABLE II
RMSE × 10^4 FOR THE SUCCESSFUL EXPERIMENTS IN THE CASE OF TABLE I

Signal-to-Noise Ratio (dB)	−4	0	4	8	12	16	20
MUSIC (forward–backward)	108	109	70	34	33	16	9
MUSIC (backward)	96	94	85	65	46	25	15
DFT-MUSIC (norm)	108	98	82	71	48	28	20
DFT-MUSIC (periodogram)	103	90	79	68	44	25	13
ACM-MUSIC (norm)	134	110	81	51	32	23	11
ACM-MUSIC (periodogram)	107	92	62	42	26	15	9

TABLE III
RMSE × 10^3 FOR ALL THE EXPERIMENTS IN THE CASE OF TABLE I

Signal-to-Noise Ratio (dB)	8	12	16	20
MUSIC (forward–backward)	82	23	1.6	0.9
MUSIC (backward)	109	48	18	1.5
DFT-MUSIC (norm)	51	38	30	14
DFT-MUSIC (periodogram)	56	39	18	7.8
ACM-MUSIC (norm)	32	20	8.0	1.1
ACM-MUSIC (periodogram)	29	19	3.7	0.9

used in evaluating (5) or (11) was $\Delta f \approx 0.0005$. The signal-to-noise ratio varied from −4 to 20 dB.

The last three rows in the tables give results for some variations of the basic approximation method to be described in Section V-D.

Table I shows the number of experiments considered successful, i.e., those in which each method could, in practice, resolve the two sinusoids. The exact condition for this is that the root-mean-square error (RMSE) $\sqrt{0.5[(f_1 - \hat{f}_1)^2 + (f_2 - \hat{f}_2)^2]}$ of the estimated frequencies $\hat{f}_1, \hat{f}_2$ [the two highest peaks of (5) or (11)] is at most 0.02.

Tables II and III give the root-mean-square error for the successful and all the 100 experiments. Because of many failures, the RMSE values for all the experiments are not very meaningful at lower SNR's and have therefore been omitted.

In Table IV, the dimensionality $M = 3$ of the signal subspace was purposely taken too large. The table shows the average number of peaks in each frequency estimator. The peak was here defined as a maximum whose value was at most −30 dB below the global maximum.

Table I shows clearly that DFT-MUSIC is a high-resolution method. It performs somewhat better than standard MUSIC at low signal-to-noise ratios. Table IV shows that DFT-MUSIC produces clearly less false peaks especially at higher signal-to-noise ratios. From the tables, one can see that for this relatively general data set some variations of DFT-MUSIC, especially computationally somewhat more demanding ACM-MUSIC, actually perform better than the basic version.

The experimental results are summarized next.

V. DISCUSSION

A. *Properties of the Approximation Compared to MUSIC*

Here, we present conclusions from all the experiments and compare the properties of the DFT-MUSIC to the basic (backward) MUSIC. In certain sense, this comparison is fair since these two methods use exactly the same set of data vectors. The conclusions are as follows:

- The approximation method is consistent in the sense that the more samples are used, the better the quality of the DFT-MUSIC frequency estimator.

TABLE IV
AVERAGE NUMBER OF PEAKS IN EACH FREQUENCY ESTIMATOR. OVERESTIMATED DIMENSIONALITY $M = 3$ OF THE SIGNAL SUBSPACE. DATA: SEE TABLE I

Signal-to-Noise Ratio (dB)	−4	0	4	8	12	16	20
MUSIC (forward–backward)	2.89	2.83	2.75	2.56	2.57	2.49	2.39
MUSIC (backward)	2.88	2.85	2.84	2.76	2.79	2.71	2.61
DFT-MUSIC (norm)	2.82	2.78	2.70	2.46	2.17	2.04	2.11
DFT-MUSIC (periodogram)	2.78	2.70	2.60	2.42	2.15	1.89	1.99
ACM-MUSIC (norm)	2.85	2.80	2.49	2.20	2.20	2.17	2.07
ACM-MUSIC (periodogram)	2.79	2.69	2.34	2.02	1.91	2.02	2.02

- The frequency estimates seem to be essentially unbiased.
- Resolution of DFT-MUSIC depends clearly on the signal-to-noise ratio. However, this dependence is even stronger in MUSIC.
- DFT-MUSIC provides, in most cases, a very good approximation to MUSIC. At high signal-to-noise ratios (over 10 dB), MUSIC is generally slightly better, producing sharper peaks. At low SNR's (e.g., 0 dB), DFT-MUSIC yields superior results. A possible explanation is given below.
- DFT-MUSIC is computationally less expensive than MUSIC. This is dealt in more detail in the next subsection.
- The DFT approximation is more robust against overestimating the number of sinusoids. In fact, some overestimation often clearly improves the resolution of DFT-MUSIC while keeping the problem of spurious peaks tolerable or even negligible. In the MUSIC method, this overestimation does not help but only causes spurious frequency peaks.
- Longer data vectors provide better resolution in both the methods up to a certain point. For DFT-MUSIC, the optimal length L of the data vectors seems to be slightly less than half of the number of samples N. In the simulations, this did not depend prominently on the SNR. For MUSIC, the optimal value of L is often the same as in DFT-MUSIC. Small deviations from the optimum affect the performance only slightly.
- A drawback of DFT-MUSIC is that it is not directly applicable to sensor array processing. However, it is easy to modify the method so that it can also be used in this area. The modified version (ACM-MUSIC) is shortly introduced in Section V-D.

An obvious question that comes into the mind is: how is it possible that DFT-MUSIC yields better results than backward-MUSIC at low SNR's even though it has originally been introduced as an approximation only? Since the estimators (5) and (11) are similar in form, and both use the same data vectors, MUSIC must miss some information. Looking at the practical realization of MUSIC in Section II-A, it is seen that the principal eigenvectors of the autocorrelation matrix (3) are estimated in quite a similar way as for arbitrary data vectors. The fact that they must be linear combinations of the signal (frequency information) vectors (4) is not explicitly exploited. At high SNR's, this does not matter so much, since the signal part dominates in the estimated correlation matrix $\hat{\boldsymbol{R}}_{xx}$. But when the number of samples is small or moderate and the SNR low, $\hat{\boldsymbol{R}}_{xx}$ may be a rather poor estimate of the theoretical form (3). Consequently, the estimated principal eigenvectors $\hat{\boldsymbol{u}}_1, \cdots, \hat{\boldsymbol{u}}_M$ do not approximate well the true signal subspace. On the other hand, in DFT-MUSIC the basis vectors of the signal subspace are sought as maximum responses of the data vectors to tentative frequencies. We believe that this additional constraint in fact improves the estimation results at low SNR's and is the basic reason of better performance of DFT-MUSIC in these situations.

The different robustness of the methods against overestimation of the number M of sinusoids can be explained as follows. MUSIC is computed from the principal eigenvectors that have the well-known optimality property (e.g., [1], [7]): they span the linear subspace that fits the data vectors best in the mean-square-error sense. If the number of principal eigenvectors is chosen correctly to M, this is good. But in the case of overestimation, the optimality property is bad, since the extra principal eigenvectors are such vectors that describe the noise best. In DFT-MUSIC, overestimation means that additional vectors $\boldsymbol{v}'_{M+1}, \cdots$, are accepted to the basis of the signal subspace approximation. These additional vectors are usually not purely noise but contain some signal portion. What remains of them after orthonormalization is not known; generally not vectors $\boldsymbol{o}_{M+1}, \cdots$, that fit the noise optimally.

B. Computational Considerations

If the number J of the data vectors $\boldsymbol{x}_k$ in (10) is chosen so that it is a power of two (e.g., by taking a suitable number N of samples or adjusting the dimensionality $L = N - J + 1$ of the data vectors $\boldsymbol{x}_k$), the transformed vectors $\boldsymbol{v}_i$ can be computed effectively using fast transforms. Considering in particular the discrete Fourier transform, one first forms L sets

$$S_1 = \{x[0], x[1], \cdots, x[J-1]\}$$
$$S_2 = \{x[1], \cdots, x[J]\}$$
$$\vdots$$
$$S_L = \{x[N-J], \cdots, x[N-1]\}$$

of the J successive samples, where the set S_j contains the jth components of the data vectors $x_0, \cdots, x_{J-1}$. Then, the FFT algorithm is applied to each set separately. The transformed vectors $v_0, \cdots, v_{J-1}$ are now obtained simply by taking as the jth component of the vector v_i the $(i + 1)$th transformed value of the set S_j.

A similar procedure can be applied to other fast transforms. We note that fast algorithms exist for other values of J than merely powers of two; see, e.g., [4]. More efficient algorithms could probably be designed by utilizing the high overlapping of the sets $S_1, \cdots, S_L$. In fact, a recursive algorithm [17] exists for computing the DFT of the set S_{j+1} efficiently in terms of the DFT of the set S_j.

The use of fast transform reduces the amount of multiplications needed in (10) from LJ^2 to about $LJ \log_2 J$. Computation of the squared norms of the v_i-vectors requires LJ multiplications (and additions), and orthonormalization of the chosen M vectors using, e.g., the modified Gram–Schmidt procedure [6] takes a further ML^2 operations.

For computing the eigenvectors of a $L \times L$ matrix, there exist several well-established and studied algorithms that have a computational complexity of order $O(L^3)$ [6]. If only a small subset of eigenvectors are needed, special algorithms can be used that reduce this amount somewhat. Before computing the eigenvectors, the correlation matrix (7) or (8) must be estimated from the available data samples. This requires about L^2K operations, even though the number of multiplications can be reduced by exploiting the special structure of the data vectors x_k.

From the above considerations, one can conclude that using the transform method to estimate the signal subspace becomes computationally especially advantageous if the dimensionality L of the data vectors is large. But just this is desirable because of improved resolution and often smaller variance.

C. Ways of Using the Approximation

Except for directly defining a MUSIC-type frequency estimator, the proposed signal subspace approximation can be used as a very good initial estimate in context with various iterative or adaptive eigenvector computation/estimation algorithms. This application has been considered in a preliminary report [10]. It turned out that even a rather crude approximation of the signal subspace based on a small number of data vectors often improved radically the convergence speed of simple LMS-type eigenvector estimation algorithms compared to using the standard random initial guess. The importance of choosing good initial estimators to the principal eigenvectors in context with such iterative methods as the power method or Lanczos method has been stressed in [24]; if the initial estimates of the eigenvectors are good, only one iteration is needed to achieve an accuracy adequate in practice.

D. Variations of the Basic Approximation Method

The choice of the M v_i-vectors defining the approximation of the signal subspace can be based on other criteria than selecting the vectors with the largest norms. An especially interesting feature of the proposed method is the possibility of using prior information for constructing the signal subspace approximation. If one has a rough knowledge of the locations of the frequencies of the sinusoids, one can directly compute the v_i-vectors corresponding to the guessed frequencies by using in (10) vectors (14) representing these frequencies as weights. The MUSIC pseudospectrum computed from the resulting signal subspace approximation is then used for extracting more accurate frequency information. For example, the periodogram can be used for obtaining such prior information by applying the technique described in [13, pp. 434–435]. We have made some experiments of this by evaluating the basis vectors at the frequencies of the M highest peaks of the data periodogram. The results for this version are given in Tables I–IV, and are actually slightly better than for the norm-based DFT-MUSIC.

A recently found promising related method is to apply the same idea of tentative frequencies to the estimated data autocorrelation matrix. More specifically, (10) is replaced by

$$v_i = \hat{R}_{xx} e_{f_i}, \qquad i = 0, 1, \cdots, L - 1. \qquad (28)$$

Here, $e_{f_0}, \cdots, e_{f_{L-1}}$ are frequency information vectors of the form (4) defined at L evenly spaced frequencies $f_0, \cdots, f_{L-1}$. This method (called ACM-MUSIC) is computationally somewhat more complex than DFT-MUSIC; but, on the other hand, it can take advantage of the forward–backward estimate (8). For comparison purposes, both the norm-based and periodogram-based version of ACM-MUSIC have been included in Tables I–IV. ACM-MUSIC performs actually best of all the methods and has similar properties with respect to the forward–backward MUSIC as DFT-MUSIC with respect to the backward MUSIC. It is applicable to sensor array processing too. The assumptions on phases are essentially the same as in MUSIC. For more results and theoretical justifications, see [11] and [12].

Instead of the DFT kernel, we have used the discrete cosine and Hadamard transform kernels in (10). The discrete cosine transform performs much the same as the DFT, but seems to be somewhat more sensitive to appearance of spurious frequency peaks when the number of sinusoids is overestimated. With real data, it may provide a MUSIC estimator that has slightly better resolution than DFT-MUSIC when the number of samples is small.

The sequency-ordered Hadamard transform [1], [7] is interesting since its kernel values are either $+1$'s or -1's, turning the multiplications in (10) to simple additions. The Hadamard basis vectors are, however, too crude approximations of frequency information vectors, and the method does not produce consistent results. Even this method yields a far better estimate of the signal subspace than a random guess and can be used successfully to initialize adaptive eigenvector estimation algorithms [10].

VI. Conclusions

In this paper, we have proposed a method for approximating the signal subspace that avoids eigenvector computation. The method is based on a generalization of discrete Fourier or other transforms to vectorial data. The approximation is generally of good quality, and can be used for defining a MUSIC-type frequency estimator. This estimator is more robust than MUSIC against noise and overestimation of the number of the sinusoids, in fact, often providing better results than MUSIC in these instances.

The most important conclusion of this study is that *it is possible to achieve a performance comparable to high-resolution eigenvector methods using the Fourier transform*, provided that the frequency estimator is formed in a similar way using a subspace formalism. Roughly speaking, Fourier transform makes the estimated signal subspace more robust while the subspace approach yields high resolution. It seems that this idea can be generalized or modified in several ways. In fact, some variations performing often better than the basic method have been briefly discussed in Section V-D.

References

[1] R. J. Clarke, *Transform Coding of Images*. New York: Academic, 1985.

[2] P. Comon and G. H. Golub, "Tracking a few extreme singular values and vectors in signal processing," *Proc. IEEE*, vol. 78, pp. 1327–1343, Aug. 1990.

[3] E. F. Deprettere, Ed., *SVD and Signal Processing*. Amsterdam, The Netherlands: North-Holland, 1988.

[4] D. F. Elliott and K. R. Rao, *Fast Transforms: Algorithms, Analyses, and Applications*. New York: Academic, 1983.

[5] J.-J. Fuchs, "Estimating the number of sinusoids in additive white noise," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 1846–1853, Dec. 1988.

[6] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore: The Johns Hopkins Press, 1983.

[7] R. C. Gonzalez and P. Wintz, *Digital Image Processing*, 2nd ed. Reading, MA: Addison-Wesley, 1987.

[8] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1986.

[9] J. Karhunen, "Recursive estimation of eigenvectors of correlation type matrices for signal processing applications," Dr. Tech. dissertation, Helsinki Univ. of Technol., Espoo, Finland, 1984.

[10] J. Karhunen, E. Nieminen, and J. Joutsensalo, "Adaptive frequency estimation by MUSIC method," Helsinki Univ. of Technol., Lab. of Comput. and Inform. Sci., Rep. A11, 1989.

[11] J. Karhunen and J. Joutsensalo, "Sinusoidal frequency estimation by approximative MUSIC method," in *Proc. 5th ASSP Workshop Spectrum Estimation Modeling*, Rochester, NY, Oct. 1990, pp. 337–341.

[12] —, "Robust MUSIC based on direct signal subspace estimation," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, Toronto, Canada, May 1991, pp. 3357–3360.

[13] S. M. Kay, *Modern Spectral Estimation: Theory and Application*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[14] S. M. Kay and A. K. Shaw, "Frequency estimation by principal component AR spectral estimation method without eigendecomposition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 95–101, Jan. 1988.

[15] S. L. Marple, Jr., *Digital Spectral Analysis with Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1987.

[16] A. N. Netravali and J. D. Limb, "Picture coding: a review," *Proc. IEEE*, vol. 68, pp. 366–406, Mar. 1980.

[17] A. Papoulis, *Signal Analysis*. New York: McGraw-Hill, 1977.

[18] B. Porat and B. Friedlander, "Analysis of the asymptotic relative efficiency of the MUSIC algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 532–544, Apr. 1988.

[19] B. D. Rao, "Perturbation analysis of an SVD-based linear prediction method for estimating the frequencies of multiple sinusoids," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 1026–1035, July 1988.

[20] R. O. Schmidt, "A signal subspace approach to multiple emitter location and spectral estimation," Ph.D. dissertation, Stanford Univ., Stanford, CA, 1981.

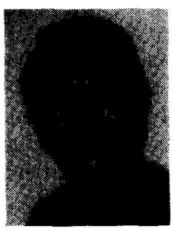
[21] —, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propagat.*, vol. 34, pp. 276–280, Mar. 1986.

[22] P. Stoica and A. Nehorai, "MUSIC, maximum likelihood, and Cramer-Rao bound," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 720–741, May 1989.

[23] D. W. Tufts and C. D. Melissinos, "Threshold extension of SVD-based algorithms," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, New York, NY, Apr. 1988, pp. 2825–2828.

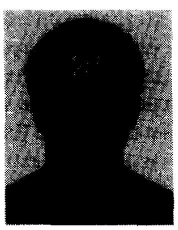
[24] —, "Simple, effective computation of principal eigenvectors and their eigenvalues and application to high-resolution estimation of frequencies," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, pp. 1046–1053, Oct. 1986.

[25] J.-F. Yang and M. Kaveh, "Adaptive eigensubspace algorithms for direction or frequency estimation and tracking," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 241–251, Feb. 1988.



Juha T. Karhunen (S'79–M'90) was born in Juva, Finland, in 1954. He received the Diploma Engineer, Licenciate of Technology, and Doctor of Technology degrees from the Helsinki University of Technology, Espoo, Finland, in 1980, 1982, and 1984, respectively.

Since 1977 he has been in various positions with the Laboratory of Computer and Information Sciences at the Helsinki University of Technology, where he is now a Senior Lecturer and Researcher. His current research interests are focused on spectral estimation and neural networks, especially eigenstructure-related methods.



Jyrki Joutsensalo was born in Kiukainen, Finland, on July 7, 1966.

Since 1988 he has been with the Laboratory of Computer and Information Sciences at the Helsinki University of Technology. His research interests are in spectral estimation and sensor array processing.