Shape from Moments—An Estimation Theory Perspective

Michael Elad, Peyman Milanfar, *Senior Member, IEEE*, and Gene H. Golub

***Abstract*—This paper discusses the problem of recovering a planar polygon from its measured complex moments. These moments correspond to an indicator function defined over the polygon's support. Previous work on this problem gave necessary and sufficient conditions for such successful recovery process and focused mainly on the case of exact measurements being given. In this paper, we extend these results and treat the same problem in the case where a longer than necessary series of noise corrupted moments is given. Similar to methods found in array processing, system identification, and signal processing, we discuss a set of possible estimation procedures that are based on the Prony and the Pencil methods, relate them one to the other, and compare them through simulations. We then present an improvement over these methods based on the direct use of the maximum–likelihood estimator, exploiting the above methods as initialization. Finally, we show how regularization and, thus, *maximum a posteriori* probability estimator could be applied to reflect prior knowledge about the recovered polygon.**

***Index Terms*—Array processing, eigenvalue, estimation, inverse problem, matrix pencil, moments, prior information, prony, quadrature, reconstruction, shape.**

I. Introduction

An intriguing inverse problem proposed in [26] suggests the reconstruction of a planar polygon from a set of its complex moments. If we consider an indicator function being 1 in the interior of the polygon and 0 elsewhere, these moments are global functions created by integrating the power function z^k over the plane and weighted by this indicator function. Given such a finite list of values, the problem posed by Milanfar *et al.* in [26] focuses on the necessary and sufficient conditions that allow a recovery of the polygon vertices from the given exact moments. In later work [11], [13], the treatment of this reconstruction problem is extended by suggesting better numerical procedures and treatment for a wider family of shapes such as algebraic curves.

When the given moments are contaminated by noise, the recovery problem becomes an estimation one. Previous work on the shape-from-moment problem concentrated on the numerical aspects of the noiseless case. In this work, we would like to extend the treatment to a given noisy but longer set of moments. The principal question we are facing is how to robustify the existing procedures to stably recover the polygon vertices from perturbed moment data.

As it turns out, the formulation of the shape-from-moments problem is very similar to several other very diverse applications found in the literature, such as i) identifying an auto-regressive system using its output; ii) decomposing a signal built as a linear mixture of complex exponentials; and iii) estimating the direction of arrival (DOA) in array processing, etc. [2], [3], [6], [15]–[18], [21]–[24], [30]–[32], [34], [35]. All these applications lead to the very same formulation and, therefore, to the same estimation problem when noise in involved. The literature in these fields offer many algorithms for solving the underlying estimation problem. It is beyond the scope of this work to survey all these results, and we will confine our exposition to a partial list of what we believe to be the leading contributions in these fields, namely, [2], [3], [6], [15]–[18], [21]–[24], [30]–[32], [34], and [35].

The contributions of this paper are twofold: First, we present a coherent overview of leading Prony- and Pencil-based algorithms in array processing, relate them to each other, put them in proper perspective, and then show their relevance to our problem. Second, we present an improvement layer over the above-mentioned methods. We start by exploring an improvement of the above algorithms by using them for the construction of an initial solution to be refined by exploiting the formulation of the problem and using directly the maximum-likelihood (ML) estimator. Through this change, we are also able to incorporate prior knowledge about the desired polygon and use a regularization term. By this, we introduce the use of the maximum *a posteriori* probability (MAP) estimator.

This paper is structured as follows: In Section II, we essentially follow [26] and formulate the shape-from-moment problem and obtain a clear relationship between the desired polygon vertices and the exact moments. In Section III, we describe a family of algorithms based on the Prony's method. Similarly, Section IV discusses a second family of algorithms based on the Pencil method. Section V presents the improvement over the above algorithms using the ML and the MAP estimation approaches. Simulations and discussion are given in Section VI. Concluding remarks are given in Section VII.

II. Problem Formulation

We start by briefly describing the formulation of the shape-from-moments reconstruction problem. This problem finds applications in diverse fields such as computed tomography, geophysical inversion, and thermal imaging. For example, in to-

Manuscript received August 9, 2002; revised July 31, 2003. This work was supported in part by the U.S. National Science Foundation under Grants CCR-9984246 and CCR-9971010. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Karim Abed-Meraim.

M. Elad is with the Computer Science Department, Technion—Israel Institute of Technology, Haifa 32000, Israel (e-mail: elad@cs.technion.ac.il).

P. Milanfar is with the Department of Electrical Engineering, School of Engineering, University of California, Santa Cruz, CA 95064-1077 USA (e-mail: milanfar@ee.ucsc.edu).

G. H. Golub is with the Department of Computer Science, Stanford University, Stanford, CA 94305-9025 USA (e-mail: golub@sccm.stanford.edu).

Digital Object Identifier 10.1109/TSP.2004.828919

mography, the X-rays of an object can be used to estimate the moments of the underlying mass distribution, and from these, the shape of the object being imaged may be estimated [11], [26]. In addition, in geophysical applications, the measurements of the exterior gravitational field of a region can be readily converted into moment information, and from these, the shape of the region may be determined [11]. In both of these examples, the framework is that of parameterized object-based geometric reconstruction, as opposed to direct pixel-based inversion of the measured data.

An arbitrary closed N-sided planar polygon P is assumed. Its vertices are denoted by $\{z_n\}_{n=1}^N$, where these values are scalar and complex. Based on Davis's Theorem [4], [5], there exists a set of N coefficients $\{a_n\}_{n=1}^N$, depending only on the vertices, such that for any analytic function $f(z)$ in the closure of P, we have

$$\int\int_P f''(z)dxdy = \sum_{n=1}^N a_n f(z_n). \tag{1}$$

Davis' Theorem shows that the coefficients $\{a_n\}_{n=1}^N$ are related to the vertices via the equation

$$a_n = \frac{i}{2}\left(\frac{\overline{z}_{n-1}-\overline{z}_n}{z_{n-1}-z_n} - \frac{\overline{z}_n-\overline{z}_{n+1}}{z_n-z_{n+1}}\right) \tag{2}$$

where $i=\sqrt{-1}$, and $\overline{z}$ is the complex conjugate of z. Note that since the polygon is closed, we define $z_{N+n}=z_n$ for all n. This formula is exploiting not only the vertices themselves, but also their connection order. For a geometric interpretation of this relationship, see [11], [26].

A special case of interest is obtained for the analytic function $f(z)=z^k$. Using (1), we get

$$\begin{aligned}\int\int_P f''(z)dxdy &= k(k-1)\int\int_P z^{k-2}dxdy \\ &= \sum_{n=1}^N a_n f(z_n) = \sum_{n=1}^N a_n z_n^k. \end{aligned}\tag{3}$$

The expression $\int\int_P z^{k-2}dxdy$ stands for the $(k-2)$th moment computed over the indicator function defined as 1 inside the polygon and zero elsewhere. We denote $k(k-1)\int\int_P z^{k-2}dxdy$ as the complex moment τ_k. Clearly, by definition, we have that $\tau_0=\tau_1=0$.

Our reconstruction problem is defined as follows: Assume that M complex moments $\{\tau_k\}_{k=0}^M$ are measured and known exactly. How can we recover the polygon vertices using the above existing relationships between these components? Note that the overall recovery problem is much more difficult if we insist on finding not only the vertices but the interior of the polygon as well, since then, we have to find the order of the vertices [7]. In this work, we concentrate on the problem of finding the vertices only. Moreover, we assume that the number of vertices N is known *a priori*.

In order to answer this question, we start by forming a set of equations from (3)

$$\begin{aligned}\forall\, k\geq 0, \tau_k = \sum_{n=1}^N a_n z_n^k \Longrightarrow & \begin{bmatrix}\tau_0\\ \tau_1\\ \tau_2\\ \vdots\\ \tau_M\end{bmatrix} \\ &= \begin{bmatrix} z_1^0 & z_2^0 & \cdots & z_N^0 \\ z_1^1 & z_2^1 & \cdots & z_N^1 \\ z_1^2 & z_2^2 & \cdots & z_N^2 \\ \vdots & \vdots & \ddots & \vdots \\ z_1^M & z_2^M & \cdots & z_N^M \end{bmatrix}\begin{bmatrix}a_1\\ a_2\\ \vdots\\ a_N\end{bmatrix}. \end{aligned}\tag{4}$$

Define $\underline{t}_{\{k_1,k_2\}}$ as the column vector of length k_2 containing the complex moments starting with τ_{k_1}. In addition, define $\mathbf{V}_{\{k_1,k_2\}}$ as the Vandermonde matrix of size $k_2\times N$ built from the vertices $\{z_n\}_{n=1}^N$ with powers starting with k_1. Finally, define the vector $\underline{a}$ as a column vector of length N containing the parameters a_n. Then, the above equation can be rewritten as

$$\underline{t}_{\{0,M+1\}} = \mathbf{V}_{\{0,M+1\}}\underline{a} \tag{5}$$

where both $\mathbf{V}_{\{0,M+1\}}$ and $\underline{a}$ are functions of the vertices. It is interesting to note that in related problems mentioned above, such as AR system identification, decomposition of a mixture of complex exponentials, and the DOA problem, a similar equation is obtained but with coefficients $\{a_n\}_{n=1}^N$ that are independent of the unknown vertices. Nevertheless, the results obtained in this paper may be applicable to these cases as well.

This equation as posed is hard to use for solving for $\{z_n\}_{n=1}^N$, given the complex moments, since it is nonlinear as z_n appear both inside the Vandermonde matrix [10] and are also hidden in the values of a_n. Moreover, solving for $\{z_n\}_{n=1}^N$, using this equation requires not only the vertices but their order as well. Alternative relations can be suggested, leading to a practical estimation procedure. These will be discussed in Section III.

To conclude the description of our problem, we have to add the perturbation (noise) issue. We assume that instead of the desired set of complex moments $\{\tau_k\}_{k=0}^M$, we have a noisy version of them $\{\hat{\tau}_k\}_{k=0}^M$, where $\hat{\tau}_k=\tau_k+u_k$, and u_k are assumed to be white i.i.d. zero-mean complex Gaussian noise with variance σ_u^2. Given these contaminated complex moments, we are interested in estimating the vertices $\{z_n\}_{n=1}^N$. We may consider, as a byproduct of the estimation process, the estimation of the clean version of the complex moments as well.

III. Prony-Based Method

A. Basic Formulation

One alternative relation to (4) can be suggested, leading to Prony's method [14], [26]. From (4), we see that $\{\tau_k\}_{k=0}^M$ satisfies an Nth-order difference equation. Thus

$$\begin{aligned}&p_N\tau_k + p_{N-1}\tau_{k+1} + \cdots \\ &\quad + p_1\tau_{k+N-1} + p_0\tau_{N+k} = 0 \text{ for} \\ &k=0,1,2,\ldots,M-N.\end{aligned}\tag{6}$$

Without loss of generality, we can assume that $p_0 = 1$ by dividing throughout by p_0. This is allowed since otherwise, the above difference equation would no longer be of Nth order.[1] Thus, we get a system of $M - N + 1$ equations of the form

$$\begin{bmatrix} p_N & \cdots & p_1 & 1 & & 0 \\ & \ddots & & \ddots & \ddots & \\ 0 & & p_N & \cdots & p_1 & 1 \end{bmatrix} \begin{bmatrix} \tau_0 \\ \tau_1 \\ \tau_2 \\ \vdots \\ \tau_M \end{bmatrix} = 0. \tag{7}$$

Simple reordering of this set of equations leads to a regular linear set of equations [11], [26]

$$-\begin{bmatrix} \tau_N \\ \tau_{N+1} \\ \tau_{N+2} \\ \vdots \\ \tau_M \end{bmatrix} = \begin{bmatrix} \tau_0 & \tau_1 & \cdots & \tau_{N-1} \\ \tau_1 & \tau_2 & \cdots & \tau_N \\ \tau_2 & \tau_3 & \cdots & \tau_{N+1} \\ \vdots & \vdots & \ddots & \vdots \\ \tau_{M-N} & \tau_{M-N+1} & \cdots & \tau_{M-1} \end{bmatrix} \times \begin{bmatrix} p_N \\ p_{N-1} \\ \vdots \\ p_1 \end{bmatrix}. \tag{8}$$

Again, for notational brevity, define $\mathbf{T}_{\{k_1,k_2,k_3\}}$ as the Hankel matrix of size $k_2 \times k_3$, built from the complex moments sequence such that the τ_{k_1} is the top left-most entry. Clearly, the matrix uses the moments $\{\tau_k\}_{k=k_1}^{k1+k2+k3-2}$. The vector $\underline{p}$ denotes the column vector consisting of the difference equation coefficients as shown above. Using this notation, the above equation is rewritten as

$$-\underline{t}_{\{N,M-N+1\}} = \mathbf{T}_{\{0,M-N+1,N\}}\underline{p}. \tag{9}$$

Note that this equation is true if the exact moments are used. Using the noisy moments, we should expect to deviate from this relationship.

In [26], it is proven that for nondegenerate polygons, the above $(M - N + 1) \times N$ matrix $\mathbf{T}_{\{0,M-N+1,N\}}$ is of full rank. Therefore, since we have $M - N + 1$ equations and N unknowns, requiring $M \geq 2N - 1$ leads to an overcomplete and well-posed system of equations.

As we will describe shortly, the difference equation coefficients $\{p_n\}_{n=1}^N$ can now be estimated using (8) in a variety of ways, such as ordinary or total least-squares. Armed with these coefficients, the vertices $\{z_n\}_{n=1}^N$ can be found by computing the roots of the polynomial

$$P(z) = \prod_{n=1}^{N}(z - z_n) = z^N + \sum_{n=1}^{N} p_n z^{N-n}. \tag{10}$$

[1]There are a variety of techniques for solving homogeneous linear set of equations, such as Gauss-elimination, QR-decomposition, and the SVD. By assigning $p_0 = 1$, the calculation is simplified, but it may not be numerically satisfactory. Later, we indeed relax this assumption.

The above polynomial equation can be obtained by applying the Z-Transform [29] to the difference equation (6).

B. Least-Squares Prony

The simplest idea for solving the above set of equations is the least-squares (LS) method of minimizing the l_2 norm of the equation's error. Then, the solution is given by [10]

$$\begin{aligned} \hat{\underline{p}} &= -\mathbf{T}^{+}_{\{0,M-N+1,N\}}\underline{t}_{\{N,M-N+1\}} \\ &= -\left(\mathbf{T}^{H}_{\{0,M-N+1,N\}}\mathbf{T}_{\{0,M-N+1,N\}}\right)^{-1} \\ &\quad \times \mathbf{T}^{H}_{\{0,M-N+1,N\}}\underline{t}_{\{N,M-N+1\}} \end{aligned} \tag{11}$$

where $\mathbf{T}^{+}$ denotes the Moore–Penrose pseudo-inverse. Given the estimated polynomial coefficients $\hat{\underline{p}}$, its roots are the estimated vertices. An efficient and stable method for finding the roots is the companion matrix method. For an $N \times N$ matrix of the form

$$\mathbf{C}(\hat{\underline{p}}) = \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 & -\hat{p}_N \\ 1 & 0 & 0 & \ldots & 0 & -\hat{p}_{N-1} \\ 0 & 1 & 0 & \ldots & 0 & -\hat{p}_{N-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 1 & -\hat{p}_1 \end{bmatrix} \tag{12}$$

its eigenvalues are the roots of the polynomial $z^N + \sum_{n=1}^{N}\hat{p}_n z^{N-n}$ [10]. This way, a root-finding problem is converted into an eigenvalue one.

C. Total-Least-Squares Prony

A total-least-squares (TLS) alternative should be preferred if we assume that error in the equations appear on both sides of (8). This corresponds to a perturbation of the complex moments sequence that introduces noise both on the entries of the vector $\underline{t}_{\{N,M-N+1\}}$, as well as on the entries of the matrix $\mathbf{T}_{\{0,M-N+1,N\}}$.

Numerically, the TLS problem is solved using the singular value decomposition (SVD) [10]. Equation (8) implies that the matrix $\mathbf{T}_{\{0,M-N+1,N+1\}}$ (note that this matrix is built with an additional column) is expected to be singular. Thus, by applying SVD on it and taking the right singular vector that corresponds to the smallest singular value, we have an alternative solution for the vector $\hat{\underline{p}}$. Note that normalization of this vector is required such that its last entry is 1; this is the coefficient of the z^N in the polynomial. Finding the vertices is accomplished, as above, by finding the roots of the polynomial, and again, the companion matrix could be used.

D. Cadzow's Iterated SVD Algorithm

Applying the SVD method as discussed above is clearly suboptimal because we are interested in a constrained SVD operation that manages not only to reduce the rank of the matrix $\mathbf{T}_{\{0,M-N+1,N+1\}}$ by one but do so while creating a new matrix having the same Hankel structure.

In his paper, Cadzow [3] suggested a simple numerical algorithm that attempts to solve this structured SVD problem. The idea is to apply an SVD rank-reduction operation by replacing the smallest singular value by zero. Since the result no longer has a Hankel structure, we need a "Hankelize" operation that averages the values along antidiagonals. Iterating between these two operations, Cadzow's algorithm converges to a local minimum of the function defined by the distance to the closest reduced-rank Hankel matrix (cf. [3]).

Cadzow's algorithm is a preprocess applied to the matrix $\mathbf{T}_{\{0,M-N+1,N+1\}}$. This matrix is constructed from the noisy complex moments, and after applying this algorithm, we may consider the new sequence as "cleaned." The left singular vector that corresponds to the smallest singular value (which is zero due to the Cadzow algorithm) leads to the estimated polynomial coefficients.

E. Other Advanced Prony-Based Methods

The methods discussed above can be considered to be approximate and inconsistent estimators [19]. They do not pose the problem as a statistical estimation one, which exploits stochastic estimators such as the maximum-likelihood [33] in order to propose an optimal solution with respect to some reasonable optimality criterion. In terms of the ML estimator, we exploit the assumption that the noise is i.i.d., Gaussian, white, and with zero mean. It is not hard to see that using the ML estimator amounts to finding the closest set of moments to the measured one, where distance is measured in the l_2-norm. This minimization should be done while forcing (8) (evaluated using the newly found moments) as a constraint.

Several successful attempts to start from this formulation of the problem and lead to a suitable numerical solution are found in the literature. Using the ML view of the problem, Bresler and Macovski suggested the iterative quadratic maximum likelihood (IQML) method [2]. Their idea is an iterative procedure that solves the total-least-squares problem with the reweighting of the noise in order to color it properly from one iteration to another. The first iteration of their method coincides exactly with the TLS method mentioned above.

Alternatively, we may regard the problem as a structured-total-least-squares one, as analyzed extensively in [6], [21]–[23], and [35]. The proposed algorithms in these works are all iterative, leading to a local minimum. The above cited analysis indicates improvement over the IQML algorithm.

Other Prony-based methods such as the modified Prony method described in [25], [27], and [28] also propose to improve the regular Prony formulation. In this work, we do not delve further into these methods, as our interest lies in the next family of Pencil-based approaches.

IV. Pencil-Based Methods

A. Basic Formulation

Starting again from (4), we now show another helpful relationship, which leads to the *pencil method* [11]. From the existing $M+1$ equations in this set, take only $M-N+1$, starting with an arbitrary index $0 \le k \le N$, and obtain

$$\begin{aligned}
\underline{t}_{\{k,M-N+1\}} &= \begin{bmatrix} \tau_k \\ \tau_{k+1} \\ \tau_{k+2} \\ \vdots \\ \tau_{M-N+k} \end{bmatrix} \\
&= \begin{bmatrix} z_1^k & \cdots & z_N^k \\ z_1^{k+1} & \cdots & z_N^{k+1} \\ z_1^{k+2} & \cdots & z_N^{k+2} \\ \vdots & \ddots & \vdots \\ z_1^{M-N+k} & \cdots & z_N^{M-N+k} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix} \\
&= \mathbf{V}_{\{k,M-N+1\}}\underline{a} = \begin{bmatrix} z_1^0 & \cdots & z_N^0 \\ z_1^1 & \cdots & z_N^1 \\ z_1^2 & \cdots & z_N^2 \\ \vdots & \ddots & \vdots \\ z_1^{M-N} & \cdots & z_N^{M-N} \end{bmatrix} \\
&\quad \times \begin{bmatrix} a_1 & & & 0 \\ & a_2 & & \\ & & \ddots & \\ 0 & & & a_N \end{bmatrix} \begin{bmatrix} z_1^k \\ z_2^k \\ \vdots \\ z_N^k \end{bmatrix} \\
&= \mathbf{V}_{\{0,M-N+1\}}\mathrm{Diag}\{\underline{a}\}\mathbf{V}_{\{k,1\}}^T. \qquad (13)
\end{aligned}$$

In the above equation, we define the operator Diag as the construction of a diagonal matrix from a given vector. By row-concatenation of the N columns corresponding to $0 \le k \le N-1$, we get

$$\begin{aligned}
\mathbf{T}_{\{0,M-N+1,N\}} &= \left[\underline{t}_{\{0,M-N+1\}}, \underline{t}_{\{1,M-N+1\}} \right. \\
&\qquad \left. \cdots \underline{t}_{\{N-1,M-N+1\}}\right] \\
&= \mathbf{V}_{\{0,M-N+1\}}\mathrm{Diag}\{\underline{a}\}\mathbf{V}_{\{0,N\}}^T. \qquad (14)
\end{aligned}$$

A similar concatenation could be built with columns corresponding to $1 \le k \le N$, resulting in

$$\begin{aligned}
\mathbf{T}_{\{1,M-N+1,N\}} &= \left[\underline{t}_{\{1,M-N+1\}}, \underline{t}_{\{2,M-N+1\}} \right. \\
&\qquad \left. \cdots \underline{t}_{\{N,M-N+1\}}\right] \\
&= \mathbf{V}_{\{0,M-N+1\}}\mathrm{Diag}\{\underline{a}\}\mathbf{V}_{\{1,N\}}^T \\
&= \mathbf{V}_{\{0,M-N+1\}} \\
&\quad \times \mathrm{Diag}\{\underline{a}\}\mathrm{Diag}\{\underline{z}\}\mathbf{V}_{\{0,N\}}^T. \qquad (15)
\end{aligned}$$

The matrix $\mathrm{Diag}\{\underline{z}\}$ is an $N \times N$ diagonal matrix with $\{z_n\}_{n=1}^N$ on its main diagonal. The square $N \times N$ Vandermonde matrix $\mathbf{V}_{\{0,N\}}$ is nonsingular since the polygon is assumed to be nondegenerate [10], [11], [26]. Based on (14) and (15), we obtain

$$\begin{aligned}
\mathbf{T}_{\{1,M-N+1,N\}}\mathbf{V}_{\{0,N\}}^{-T} &= \mathbf{V}_{\{0,M-N+1\}}\mathrm{Diag}\{\underline{a}\} \\
&\quad \times \mathrm{Diag}\{\underline{z}\}\mathbf{V}_{\{0,N\}}^T\mathbf{V}_{\{0,N\}}^{-T} \\
&= \mathbf{V}_{\{0,M-N+1\}}\mathrm{Diag}\{\underline{a}\}\mathrm{Diag}\{\underline{z}\} \\
&= \mathbf{V}_{\{0,M-N+1\}}\mathrm{Diag}\{\underline{a}\} \\
&\quad \times \mathbf{V}_{\{0,N\}}^T\mathbf{V}_{\{0,N\}}^{-T}\mathrm{Diag}\{\underline{z}\} \\
&= \mathbf{T}_{\{0,M-N+1,N\}}\mathbf{V}_{\{0,N\}}^{-T}\mathrm{Diag}\{\underline{z}\}. \\
&\qquad (16)
\end{aligned}$$

This cumbersome relationship actually implies that for the pair of matrices $\mathbf{T}_{\{1,M-N+1,N\}}$ and $\mathbf{T}_{\{0,M-N+1,N\}}$, the vertices are their generalized eigenvalues, and the columns of the matrix $\mathbf{V}_{\{0,N\}}^{-T}$ are their generalized eigenvectors. Thus, given the sequence of moments $\{\tau_k\}_{k=0}^{M}$, we are to form the two $(M-N+1)\times N$ Hankel matrices $\mathbf{T}_{\{1,M-N+1,N\}}$ and $\mathbf{T}_{\{0,M-N+1,N\}}$ and then solve for their generalized eigenvalues using the relation

$$(\mathbf{T}_{\{1,M-N+1,N\}}-\lambda\mathbf{T}_{\{0,M-N+1,N\}})\underline{v}=0. \tag{17}$$

The eigenvalues are the vertices we desire. A complicating aspect in this relationship is the fact that the obtained pencil is rectangular with more rows than columns. Note that this relationship is true for the noiseless complex moments, and even weak noise added to these moments may lead to no solution for this system of equations.

B. Intuitive Pencil Averaging

One natural and completely heuristic measure to cope with such rectangular pencils is taking only a square slice of it using only N rows. Clearly, for the noiseless moments, this should work well. Then, given the created square pencil, a regular generalized eigenvalue solver, such as the QZ decomposition algorithm [10], can be used. The found eigenvalues are our estimated vertices.

Since, for the pencil in (17), there are many N subgroups of rows, we may consider picking several of them, computing the eigenvalues, and averaging the results. In particular, we may sweep with a sliding window of length N through the $M-N+1$ rows and use the $M-2N+2$ squares systems. Note that a difficulty of the latter approach is the need to correspond the eigenvalues one to another between different solutions. This could be done by some clustering method. Our experiments indicates that this method is very sensitive and does not show promise.

C. Squaring the Pencil

If we could square the pencil to an $N\times N$ one, we could have proposed a solution. A sophisticated, yet presumably arbitrary squaring method, is multiplying both sides of (17) with either $\mathbf{T}_{\{0,M-N+1,N\}}^{H}$ or $\mathbf{T}_{\{0,M-N+1,N\}}^{+}$. This of course will square the pencil as desired. Can we justify such an operation?

As it turns out, one could prove an exact equivalence between both these squaring methods and the employment of the Prony's method based on least-squares [15], [16]. Let us briefly show this result, as it sheds light on the Pencil method and its claimed superiority over the Prony's approach.

Based on the prony-LS solution, as given in (11), we may write the following extended equation:

$$\begin{aligned}\mathbf{T}_{\{0,M-N+1,N\}}^{+}&\mathbf{T}_{\{0,M-N+1,N+1\}}\\ &=\mathbf{T}_{\{0,M-N+1,N\}}^{+}\left[\mathbf{T}_{\{0,M-N+1,N\}},\ \underline{t}_{\{N,M-N+1\}}\right]\\ &=\left[\mathbf{I}_N,-\underline{\hat{p}}\right]=\begin{bmatrix}1&0&0&\dots&0&-\hat{p}_N\\0&1&0&\dots&0&-\hat{p}_{N-1}\\0&0&0&\dots&0&-\hat{p}_{N-2}\\\vdots&\vdots&\vdots&\ddots&\vdots&\vdots\\0&0&0&\dots&1&-\hat{p}_1\end{bmatrix}.\end{aligned} \tag{18}$$

In this result, the N right columns exactly form the companion matrix $\mathbf{C}(\underline{\hat{p}})$, as described in (12). Therefore, we have immediately that

$$\mathbf{T}_{\{0,M-N+1,N\}}^{+}\mathbf{T}_{\{1,M-N+1,N+1\}}=\mathbf{C}(\underline{\hat{p}}). \tag{19}$$

Returning to the squaring method discussed here, assume that we have multiplied both sides of (17) with $\mathbf{T}_{\{0,M-N+1,N\}}^{H}$, getting

$$\begin{aligned}\mathbf{T}_{\{0,M-N+1,N\}}^{H}&\mathbf{T}_{\{1,M-N+1,N\}}\underline{v}\\ &=\lambda\mathbf{T}_{\{0,M-N+1,N\}}^{H}\mathbf{T}_{\{0,M-N+1,N\}}\underline{v}.\end{aligned} \tag{20}$$

A somewhat unstable solution of the obtained generalized eigenvalue problem is obtained by multiplying both sides by the inverse of the right matrix, resulting in

$$\begin{aligned}&\left(\mathbf{T}_{\{0,M-N+1,N\}}^{H}\mathbf{T}_{\{0,M-N+1,N\}}\right)^{-1}\\ &\quad\times\mathbf{T}_{\{0,M-N+1,N\}}^{H}\mathbf{T}_{\{1,M-N+1,N\}}\underline{v}\\ &\qquad=\mathbf{T}_{\{0,M-N+1,N\}}^{+}\mathbf{T}_{\{1,M-N+1,N\}}\underline{v}=\lambda\underline{v}.\end{aligned} \tag{21}$$

In this way, we find that the squaring method of the pencil leads to the need to find the eigenvalues of the matrix $\mathbf{T}_{\{0,M-N+1,N\}}^{+}\mathbf{T}_{\{1,M-N+1,N\}}$, and as we have shown above, this is exactly the companion matrix used by the LS-Prony method. Therefore, we see a close relation between the proposed squaring method and the LS-Prony method. A similar equivalence could be shown for squaring using $\mathbf{T}_{\{0,M-N+1,N\}}^{+}$ instead of $\mathbf{T}_{\{0,M-N+1,N\}}^{H}$.

Note that in favor of the pencil-squaring method, we should mention that solving a square generalized eigenvalue problem $(\mathbf{A}-\lambda\mathbf{B})\underline{v}=0$ by converting it to a regular eigenvalue one of the form $(\mathbf{B})^{-1}\mathbf{A}-\lambda\mathbf{I}\underline{v}=0$ is considered to be a crude and poor numerical method. Thus, the pencil-squaring approach is expected to give better accuracy. However, since it is closely related to the Prony-LS method, we see its suboptimality since it disregards the inherent Hankel structure. On the other hand, in favor of the LS-Prony method, we should say that it is specifically using the knowledge that some of the entries in the matrix given in (18) are exact zeros. The pencil-squaring method does not use this knowledge; hence, it does not exploit the $N-1$ columns overlap between the two involved matrices.

D. "Generalized Pencil of Function" Method

The method "generalized pencil of function" (GPOF), which has been promoted by Hua and Sarkar [15]–[18], is a successful attempt to improve on the above squaring method. It leads again to a square pencil that can be dealt with in the usual way. This squaring method, however, is somewhat more complicated than the one discussed in Section IV-C. A perturbation analysis done on this method indicates results close to those predicted by the Cramér–Rao bound [2], [33], which basically means a near-optimal solution. In [17], a relationship between this method and several variants of the ESPRIT method [31], [32] is derived showing comparable performance. Later work described in [24], [30], and [34] further improved the GPOF results by forcing the Hankel structure based on the Cadzow procedure or by easing the algorithm's complexity. In what follows, we will confine our presentation to the core GPOF idea.

When forming the pencil in (17), we gathered exactly N columns per both $\mathbf{T}_{\{0,M-N+1,N\}}$ and $\mathbf{T}_{\{1,M-N+1,N\}}$. What happens if we gather more? Returning to (14) and (15), assume that $L \geq N$ columns are gathered using the columns $\underline{t}_{\{k,*\}}$ with $0 \leq k \leq L-1$. Since we have a finite moments set $\{\tau_k\}_{k=0}^{M}$ at our disposal, we must use fewer rows so that the new matrix pair exploits only the existing moments. Originally, $M-N+1$ rows were used, and now, this should change to $M-L+1$. Thus

$$\begin{aligned}\mathbf{T}_{\{0,M-L+1,L\}} &= \Big[\underline{t}_{\{0,M-L+1\}}, \underline{t}_{\{1,M-L+1\}} \\ &\qquad \cdots \underline{t}_{\{L-1,M-L+1\}}\Big] \\ &= \mathbf{V}_{\{0,M-L+1\}}\text{Diag}\{\underline{a}\}\mathbf{V}_{\{0,L\}}^{T}. \quad (22)\end{aligned}$$

This matrix uses $\{\tau_k\}_{k=0}^{M-1}$ in its entries. Its size is $(M-L+1)\times L$, but its rank is N due to the $N\times N$ diagonal matrix in this decomposition. Note that L is bounded from above by $L \leq M-N+1$, as any value above this limit results with a matrix with fewer rows or columns and, thus, lower rank than N. A similar concatenation corresponding to the columns $1 \leq k \leq L$ results in

$$\begin{aligned}\mathbf{T}_{\{1,M-L+1,L\}} &= \Big[\underline{t}_{\{1,M-L+1\}}, \underline{t}_{\{2,M-L+1\}} \\ &\qquad \cdots \underline{t}_{\{L,M-L+1\}}\Big] \\ &= \mathbf{V}_{\{0,M-L+1\}}\text{Diag}\{\underline{a}\}\text{Diag}\{\underline{z}\}\mathbf{V}_{\{0,L\}}^{T}. \quad (23)\end{aligned}$$

The Vandermonde matrix $\mathbf{V}_{\{0,L\}}^{T}$ has N rows and $L \geq N$ columns, and due to the nondegenerate polygon, it is nonsingular and of rank N. We now show that these two matrices still satisfy a pencil relationship since for $L \geq N$, we have

$$\begin{aligned}&\mathbf{T}_{\{1,M-L+1,L\}}\mathbf{V}_{\{0,L\}}\left(\mathbf{V}_{\{0,L\}}^{T}\mathbf{V}_{\{0,L\}}\right)^{-1} \\ &\quad = \mathbf{V}_{\{0,M-L+1\}}\text{Diag}\{\underline{a}\}\text{Diag}\{\underline{z}\} \\ &\quad = \mathbf{V}_{\{0,M-L+1\}}\text{Diag}\{\underline{a}\}\mathbf{V}_{\{0,L\}}^{T}\mathbf{V}_{\{0,L\}} \\ &\qquad \times \left(\mathbf{V}_{\{0,L\}}^{T}\mathbf{V}_{\{0,L\}}\right)^{-1}\text{Diag}\{\underline{z}\} \\ &\quad = \mathbf{T}_{\{0,M-L+1,L\}}\mathbf{V}_{\{0,L\}} \\ &\qquad \times \left(\mathbf{V}_{\{0,L\}}^{T}\mathbf{V}_{\{0,L\}}\right)^{-1}\text{Diag}\{\underline{z}\}. \quad (24)\end{aligned}$$

Here, we have used the fact that $\left(\mathbf{V}_{\{0,L\}}^{T}\mathbf{V}_{\{0,L\}}\right)$ is a square $N\times N$ full rank matrix, and thus, it is invertible. For a new pair of $(M-L+1)\times L$ matrices, we got a pencil just as before, with eigenvectors being the columns of the matrix $\mathbf{V}_{\{0,L\}}\left(\mathbf{V}_{\{0,L\}}^{T}\mathbf{V}_{\{0,L\}}\right)^{-1}$. To summarize, we require that the following generalized eigenvalue problem be solved:

$$\left(\mathbf{T}_{\{1,M-L+1,L\}} - \lambda\mathbf{T}_{\{0,M-L+1,L\}}\right)\underline{v} = 0 \quad (25)$$

and this equation is a generalization of (17); when $L=N$, these equations coincide.

Now, assume that we follow the squaring idea mentioned above, and multiply both sides of this equations by $\mathbf{T}_{\{0,M-L+1,L\}}^{+}$. This way, we get an $L\times L$ solvable pencil of the form

$$\begin{aligned}&\Big(\mathbf{T}_{\{0,M-L+1,L\}}^{+}\mathbf{T}_{\{1,M-L+1,L\}} \\ &\qquad -\lambda\mathbf{T}_{\{0,M-L+1,L\}}^{+}\mathbf{T}_{\{0,M-L+1,L\}}\Big)\underline{v} = 0. \quad (26)\end{aligned}$$

An interesting property that can be exploited here is the fact that the two $L\times L$ matrices $\mathbf{T}_{\{0,M-L+1,L\}}^{+}\mathbf{T}_{\{0,M-L+1,L\}}$ and $\mathbf{T}_{\{0,M-L+1,L\}}^{+}\mathbf{T}_{\{1,M-L+1,L\}}$ share the same $L-N$-dimensional null space. As shown in (22) and (23), both of these matrices are created by multiplication with the rank N matrix $\mathbf{V}_{\{0,L\}}$ on the right. Therefore, only N of this pencil's eigenvalues correspond to the desired vertices, and the remaining $L-N$ refer to this common null space. A basis that spans this null space and its orthogonal complement can be obtained by performing an SVD decomposition of the form $\mathbf{T}_{\{0,M-L+1,L\}} = \mathbf{U}\mathbf{D}\mathbf{Z}^{H}$. Then, defining $\mathbf{Z}_1$ as the first N columns of $\mathbf{Z}$ and $\mathbf{Z}_2$ as the last $L-N$ ones, it is clear that $\mathbf{Z}_2$ spans the pencil's null space.

Thus far, we managed to square the pencil to an $L\times L$ one. We still have to shrink it to get to an $N\times N$ pencil that leads to the N vertices as eigenvalues. One step toward this goal is the assumption that the eigenvectors to consider are to be orthonormal to the null space of the pencil. Thus, the candidate eigenvectors for the pencil should be spanned by $\mathbf{Z}_1$, and this could be written as

$$\begin{aligned}0 &= \Big(\mathbf{T}_{\{0,M-L+1,L\}}^{+}\mathbf{T}_{\{1,M-L+1,L\}} \\ &\qquad -\lambda\mathbf{T}_{\{0,M-L+1,L\}}^{+}\mathbf{T}_{\{0,M-L+1,L\}}\Big)\mathbf{Z}_1\underline{\alpha} \\ &= \left(\mathbf{Z}^{H}\mathbf{D}^{+}\mathbf{U}^{H}\mathbf{T}_{\{1,M-L+1,L\}}\mathbf{Z}_1 \right. \\ &\qquad \left. -\lambda\mathbf{Z}^{H}\mathbf{D}^{+}\mathbf{U}^{H}\mathbf{U}\mathbf{D}\mathbf{Z}\mathbf{Z}_1\right)\underline{\alpha} \\ &= \mathbf{Z}^{H}\left(\mathbf{D}^{+}\mathbf{U}^{H}\mathbf{T}_{\{1,M-L+1,L\}}\mathbf{Z}_1 - \lambda\mathbf{D}^{+}\mathbf{D}\mathbf{Z}\mathbf{Z}_1\right)\underline{\alpha}. \quad (27)\end{aligned}$$

This operation turns our pencil size to become $L\times N$. The multiplication by the square and unitary $L\times L$ matrix $\mathbf{Z}$ on the left does not change the pencil's solution and, thus, can be discarded. Since the diagonal matrix $\mathbf{D}$ is expected to have N nonzero first entries along its main diagonal and zeros elsewhere, it is clear that the last $L-N$ rows of this pencil are expected to be zeros. Thus, defining $\mathbf{D}_1$ to be the upper left $N\times N$ part of $\mathbf{D}$ and $\mathbf{U}_1$ to be the left N columns of $\mathbf{U}$, we obtain

$$0 = \left(\mathbf{D}_1^{-1}\mathbf{U}_1^{H}\mathbf{T}_{\{1,M-L+1,L\}}\mathbf{Z}_1 - \lambda\mathbf{I}\right)\underline{\alpha}. \quad (28)$$

This is a square $N\times N$ pencil, as desired, and its solution leads to N estimated vertices.

V. Improved Estimation Algorithm

A. Exact ML Refinement

Let us return now to the basic relation we saw in (4) that states $\underline{t}_{\{0,M+1\}} = \mathbf{V}_{\{0,M+1\}}\underline{a}$. Note that in this equation, both the matrix $\mathbf{V}_{\{0,M+1\}}$ and vector $\underline{a}$ are functions of the vertices. If the measured moments are contaminated with white Gaussian

noise with variance σ_u^2, then the ML estimate of the vertices $\{z_n\}_{n=1}^N$ is obtained by

$$\begin{aligned}\{\hat{z}_n\}_{n=1}^N &= \text{ArgMax}_{z_1,z_2,\ldots z_N} \text{PDF}\left\{\underline{t}_{\{0,M+1\}} | \{z_n\}_{n=1}^N\right\} \\ &= \text{ArgMax}_{z_1,z_2,\ldots z_N} \frac{1}{(2\pi\sigma_u^2)^{N/2}} \\ &\quad \cdot \exp\left\{-\frac{\|\underline{t}_{\{0,M+1\}} - \mathbf{V}_{\{0,M+1\}}\underline{a}\|^2}{2\sigma_u^2}\right\} \\ &= \text{ArgMin}_{z_1,z_2,\ldots z_N} \|\underline{t}_{\{0,M+1\}} - \mathbf{V}_{\{0,M+1\}}\underline{a}\|^2 \\ &= \text{ArgMin}_{z_1,z_2,\ldots z_N} \sum_{k=0}^{M}\left|\tau_k \right. \\ &\quad \left. - \sum_{n=1}^{N} \frac{i}{2}\left(\frac{\overline{z}_{n-1}-\overline{z}_n}{z_{n-1}-z_n} - \frac{\overline{z}_n - \overline{z}_{n+1}}{z_n - z_{n+1}}\right) z_n^k\right|^2 \quad (29)\end{aligned}$$

where PDF denotes the probability density function of the argument. In the above, we have used (2). As we have already said before, using this minimization problem to directly solve for the unknown vertices leads to two difficulties: i) Unless we successfully initialize the optimization procedure, we are bound to fall into a local minimum; and ii) using this expression calls for the need to solve the problem of ordering the vertices.

As to the first problem, we can assume that one of the above mentioned estimation (either Prony- or Pencil-based) methods is used, and a reasonable estimate of the polygon vertices is indeed given. Thus, using this solution for initial values, we can expect to improve when minimizing, even locally, the above function. This implies that the new method is more robust than previously discussed algorithms. Since the exact ML method directly formulates the desired goal in our estimation task without assumptions or approximations, it is expected to lead to better results compared with the algebraic methods. These methods, while mathematically more appealing, neglect the dependencies of the coefficients on the vertices, not to mention the statistical optimality condition provided by the ML solution.

For the problem of ordering, we may consider two options: i) Assume that we are able to order the vertices given to us; see [7] for discussion on how to solve this problem, or ii) disregard the dependency of the $\underline{a}$ coefficients on the vertices, and replace this vector with the least-squares minimizer of this error, namely

$$\begin{aligned}\underline{a} &= \left[\mathbf{V}_{\{0,M+1\}}^H \mathbf{V}_{\{0,M+1\}}\right]^{-1} \mathbf{V}_{\{0,M+1\}}^H \underline{t}_{\{0,M+1\}} \\ &= \mathbf{V}_{\{0,M+1\}}^{+} \underline{t}_{\{0,M+1\}}. \quad (30)\end{aligned}$$

This way, we ignore the relation between the unknown vertices and the coefficients $\underline{a}$ (and thus, we necessarily lose something), but we gain from the simplicity induced by the freedom of finding the order of the vertices. Actually, the second approach is suitable for using the proposed idea on improvement when dealing with applications such as AR-system identification, and more, as indeed, $\underline{a}$ is not a function of the unknowns in any direct way.

Let us define our target function to be minimized by

$$\begin{aligned}&f(z_1, z_2, \ldots, z_N) \\ &= \sum_{k=0}^{M}\left|\tau_k - \sum_{n=1}^{N} \frac{i}{2}\left(\frac{\overline{z}_{n-1}-\overline{z}_n}{z_{n-1}-z_n} - \frac{\overline{z}_n-\overline{z}_{n+1}}{z_n-z_{n+1}}\right) z_n^k\right|^2. \quad (31)\end{aligned}$$

Let us assume that given an initial solution, we are able to order the vertices properly. Since this function is complicated, we propose to exploit a numerical algorithm in order to fine tune this initial solution and, in this way obtain the desired improvement. One such simple idea is to perform a line search for each vertex with all the other points fixed. In this way, we update the algorithm through a coordinate descent optimization procedure. Alternatively, more sophisticated nonlinear least-squares methods could be used.

It is interesting to note that there is a close relationship between the method proposed here and the variable-projection (VarPro) method [8] and its variants [12], [20]. Starting with the assumption that the coefficients $\{a_n\}_{n=1}^N$ are independent of the vertices $\{z_n\}_{n=1}^N$, the above minimization problem (31) becomes

$$\text{ArgMin}_{z_1,z_2,\ldots z_N} \sum_{k=0}^{M}\left|\tau_k - \sum_{n=1}^{N} a_n z_n^k\right|^2.$$

This problem is treated by the VarPro method, as originally proposed by Golub and Pereyra [8]. VarPro proposes to directly minimize the function obtained by the ML formulation instead of using a simplified formulation such as the Prony or the Pencil methods. VarPro exploits the fact that the variables of the optimization problem are separable and that the problem is linear with respect to the $\{a_n\}_{n=1}^N$. Thus, an explicit expression for these coefficients can be obtained by ordinary least-squares and used in the penalty function shown above. As to the remaining unknowns, these are found iteratively, exploiting closed-form expressions for the derivatives of the function.

As was indicated before, disregarding the relationship between the $\{a_n\}_{n=1}^N$ and the $\{z_n\}_{n=1}^N$ may reduce the accuracy of the obtained solution. A variant of the VarPro proposed by Kaufman and Pereyra [20] is more suitable for our problem, as it allows for the introduction of constraints as well. Our problem thus can be rewritten as

$$\begin{aligned}\text{ArgMin}_{z_1,z_2,\ldots z_N} \quad & \sum_{k=0}^{M}\left|\tau_k - \sum_{n=1}^{N} a_n z_n^k\right|^2 \\ \text{Subject To:} \quad & \left\{a_n = \frac{i}{2}\left(\frac{\overline{z}_{n-1}-\overline{z}_n}{z_{n-1}-z_n} \right.\right. \\ & \left.\left. - \frac{\overline{z}_n-\overline{z}_{n+1}}{z_n-z_{n+1}}\right)\right\}_{n=1}^{N}.\end{aligned}$$

Indeed, using VarPro with constraints as in [20], we get an algorithm to minimize the ML penalty function we have defined. The differences between this approach and ours are minor, namely, the choice of initialization, and the fact that our algorithm employs a derivative-free optimization procedure.

B. Regularization and the MAP Estimator

From an intuitive point of view, we can suggest the following: If we have some prior knowledge about the desired vertices, we can exploit this information and direct the result toward this property by adding a regularization term to (31). As an example,

knowing that the angles formed by the vertices are close to 90°, we may add a term of the form

$$\text{Reg}\left(\{z_n\}_{n=1}^N\right) = \sum_{n=1}^{N}\left(\left|\frac{1}{2}\left(\frac{\overline{z}_{n-1}-\overline{z}_n}{z_{n-1}-z_n}-\frac{\overline{z}_n-\overline{z}_{n+1}}{z_n-z_{n+1}}\right)\right|-1\right)^2.$$

This expression exploits the geometric interpretation of the a_n coefficients, having a unit magnitude for vertices forming 90° angle. Since the minimization described above is done numerically, any reasonable regularization function can be incorporated and used. When adding this term, we should multiply it by some confidence factor λ. Large λ implies that we are confident about this property of the vertices, and thus, this penalty should play a stronger role. An automatic choice of λ can also be made based on, for instance, the generalized cross-validation (GCV) method [9].

There are many other choices for the regularization function. Just to mention a few, one might be interested in smoothness of the final polygon, suggesting

$$\text{Reg}\left(\{z_n\}_{n=1}^N\right) = \sum_{n=1}^{N}|z_n - z_{n-1}|^2$$

or

$$\text{Reg}\left(\{z_n\}_{n=1}^N\right) = \sum_{n=1}^{N}|2z_n - z_{n-1} - z_{n+1}|^2.$$

Alternatively, we might direct the solution to a less "rough" polygon using the fact that the polygon area is given by $\mathcal{A} = 0.5\text{Im}\left\{\sum_{n=1}^{N}\overline{z}_n z_{n+1}\right\}$ [see (3)], and the perimeter is $\Pi = \sum_{n=1}^{N}|z_n - z_{n-1}|$. Defining $\text{Reg}\left(\{z_n\}_{n=1}^N\right) = \Pi^2/\mathcal{A}$ or $\text{Reg}\left(\{z_n\}_{n=1}^N\right) = \Pi^2 - 4\pi\mathcal{A}$, we can measure "roughness" and penalize for it.

This regularization idea essentially leads to the maximum *a posteriori* probability (MAP) estimator. The MAP estimator maximizes the posterior probability of the unknowns given the measurements $\text{PDF}\left\{\{\hat{z}_n\}_{n=1}^N \mid \underline{t}_{\{0,M+1\}}\right\}$, and using Bayes rule, this is given by

$$\begin{aligned}\{\hat{z}_n\}_{n=1}^N &= \text{ArgMax}_{z_1,z_2,\ldots z_N}\text{PDF}\left\{\{\hat{z}_n\}_{n=1}^N \mid \underline{t}_{\{0,M+1\}}\right\}\\ &= \text{ArgMax}_{z_1,z_2,\ldots z_N}\text{PDF}\left\{\underline{t}_{\{0,M+1\}} \mid \{\hat{z}_n\}_{n=1}^N\right\}\\ &\quad\cdot\text{PDF}\left\{\{\hat{z}_n\}_{n=1}^N\right\}. \qquad (32)\end{aligned}$$

If we consider the same likelihood function as in (29) and further assume that

$$\text{PDF}\left\{\{z_n\}_{n=1}^N\right\} = C\cdot\exp\left\{-\text{Reg}\left(\{z_n\}_{n=1}^N\right)\right\} \qquad (33)$$

where C is a normalization coefficient, and the function $\text{Reg}(*)$ is some energy function, then we get that our overall target function becomes

$$\frac{f(z_1,z_2,\ldots,z_N)}{2\sigma_u^2} + \text{Reg}\left(\{z_n\}_{n=1}^N\right) \qquad (34)$$

and this is indeed the regularization approach of interest.

VI. Results

In this section, we present reconstruction results corresponding to some of the algorithms presented in this paper.

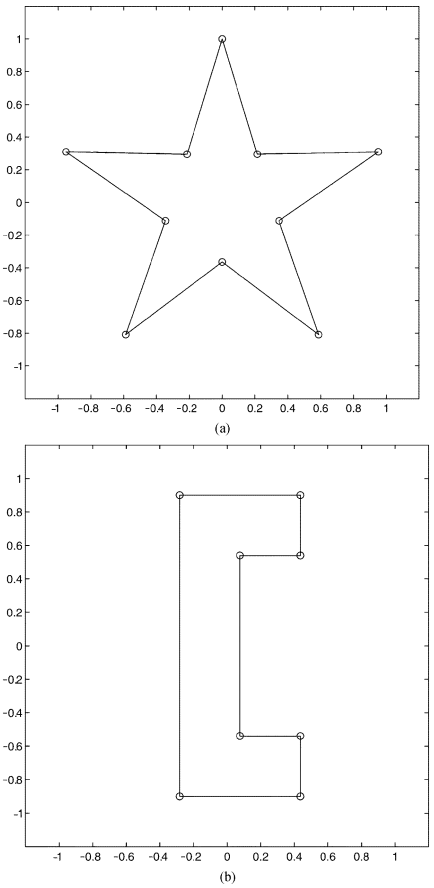


Fig. 1. Polygons used for the described experiments.

The structure of the experiment is as follows: We start by creating a polygon and computing its complex moments using Davis's Theorem (2) and (4). We then add complex Gaussian white noise to the moments and apply several of the estimation procedures discussed above. Several comments are in order before we proceed.

- In creating the polygon, we normalize it to be centered around the origin and lie inside the unit disk to obtain a stable moment series. This operation could be interpreted as adjusting our measurement devices to cope with exponentially growing/shrinking measurements. More on this normalization effect can be found in [11] and [13].
- The noise added in the experiment is relatively weak. We found that *all* the algorithms essentially fail when the noise energy is above some threshold. This hints at the complexity of the problem and its severe inherent sensitivity. Note that regularization does improve the stability of these algorithms to some extent.
- Each experiment is repeated 100 times (unless otherwise stated) in order to average error results. We denote the estimated vertices in the jth experiment by $\{\hat{z}_n(j)\}_{n=1}^N$ and the exact locations as $\{z_n\}_{n=1}^N$. Thus, the root mean squared error per vertex is given by

$$\sqrt{\frac{1}{100}\sum_{j=1}^{100}|\hat{z}_n(j) - z_n|^2}.$$

TABLE I
STAR SHAPE WITH NOISE FACTOR $\sigma_u - 1e-3$—RMSE PER VERTEX

Method	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8	# 9	# 10
LS-Prony	1.9e-5	8.0e-2	1.9e-5	8.5e-2	1.8e-5	**7.9e-2**	1.8e-5	8.7e-2	2.0e-5	8.6e-2
TLS-Prony	3.5e-5	4.6e-1	4.1e-5	4.8e-1	2.8e-5	4.1e-1	3.7e-5	4.9e-1	3.6e-5	4.5e-1
Cadzow	2.0e-5	4.8e-1	3.5e-5	5.4e-1	2.1e-5	4.9e-1	2.2e-5	5.5e-1	6.1e-5	5.4e-1
Pencil Ave.	3.8e-5	4.9e-1	3.7e-5	5.1e-1	3.3e-5	5.0e-1	3.7e-5	5.1e-1	3.8e-5	4.9e-1
Pencil Squaring	1.7e-5	**7.9e-2**	1.8e-5	**8.3e-2**	2.0e-5	8.3e-2	1.9e-5	**8.1e-2**	1.9e-5	**8.1e-2**
GPOF method	**1.1e-5**	5.1e-1	**1.1e-5**	6.5e-1	**1.2e-5**	4.9e-1	**1.2e-5**	5.5e-1	**1.1e-5**	6.0e-1

TABLE II
STAR SHAPE WITH NOISE FACTOR $\sigma_u = 1e-4$—RMSE PER VERTEX

Method	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8	# 9	# 10
LS-Prony	2.0e-6	2.8e-2	1.8e-6	2.8e-2	1.9e-6	3.0e-2	1.7e-6	2.8e-2	1.9e-6	2.9e-2
TLS-Prony	2.1e-6	3.4e-2	1.9e-6	3.3e-2	2.0e-6	3.6e-2	1.8e-6	3.4e-2	2.0e-6	3.5e-2
Cadzow	1.6e-6	2.9e-2	1.5e-6	2.9e-2	1.4e-6	3.1e-2	1.3e-6	2.9e-2	1.4e-6	3.0e-2
Pencil Ave.	4.2e-6	4.9e-1	3.5e-6	4.8e-1	4.4e-6	4.8e-1	4.1e-6	4.9e-1	3.6e-6	4.8e-1
Pencil Squaring	1.8e-6	2.8e-2	2.0e-6	2.8e-2	1.9e-6	2.6e-2	1.9e-6	2.9e-2	1.9e-6	2.7e-2
GPOF method	**1.2e-6**	**2.5e-2**	**1.1e-6**	**2.5e-2**	**1.2e-6**	**2.5e-2**	**1.2e-6**	**2.4e-2**	**1.3e-6**	**2.5e-2**

TABLE III
STAR SHAPE WITH NOISE FACTOR $\sigma_u = 1e-5$—RMSE PER VERTEX

Method	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8	# 9	# 10
LS-Prony	1.9e-7	3.1e-3	2.0e-7	3.1e-3	1.7e-7	3.0e-3	1.8e-7	3.0e-3	2.0e-7	3.0e-3
TLS-Prony	1.9e-7	3.1e-3	2.0e-7	3.1e-3	1.7e-7	3.0e-3	1.8e-7	3.0e-3	2.0e-7	3.0e-3
Cadzow	1.3e-7	2.7e-3	1.4e-7	2.7e-3	1.3e-7	2.7e-3	1.5e-7	2.7e-3	1.5e-7	2.7e-3
Pencil Ave.	3.8e-7	4.7e-1	4.3e-7	4.8e-1	4.0e-7	4.8e-1	1.1e-6	4.7e-1	5.1e-7	4.8e-1
Pencil Squaring	1.8e-7	2.8e-3	1.8e-7	3.0e-3	1.8e-7	3.0e-3	1.9e-7	3.0e-3	2.0e-7	2.9e-3
GPOF method	**1.2e-7**	**2.5e-3**	**1.2e-7**	**2.5e-3**	**1.2e-7**	**2.4e-3**	**1.2e-7**	**2.5e-3**	**1.2e-7**	**2.6e-3**

TABLE IV
STAR SHAPE—OVERALL RMSE ERROR PER METHOD

Noise variance	LS-Prony	TLS-Prony	Cadzow Alg.	Pencil Average	Pencil Squar.	GPOF method
1e-3	5.90e-2	3.23e-1	3.68e-1	3.68e-1	**5.74e-2**	3.99e-1
1e-4	2.01e-2	2.43e-2	2.10e-2	3.42e-1	1.96e-2	**1.74e-2**
1e-5	2.20e-3	2.21e-3	1.91e-3	3.36e-1	2.11e-3	**1.71e-3**

RMSE for all the vertices is obtained by averaging these values. Since all the vertices are in the unit circle, by simply estimating the vertices as being on the origin, we obtain an error of less than 1. This should be used as a reference value in order to interpret the presented values. For example, when a method is shown to give an average error of 0.5, it really means that this method fails completely.

- In our simulation, we use the least-squares Prony method, the TLS-Prony method, the Cadzow algorithm with ten iterations, the pencil averaging method, the squaring of the nonsquare pencil by the multiplication by $\mathbf{T}^H_{\{0,M-N+1,N\}}$, and the GPOF method with $L = M/2$. In all these methods, we start by nulling τ_0, τ_1, and τ_3 to zero. The fact that $\tau_0 = \tau_1 = 0$ has been indicated by (3). In addition, $\tau_3 = 0$ since it is proportional to the center of mass, and this is zero due to the normalization mentioned above.

Experiment 1: In this experiment, we use a star-shape with ten vertices, as shown in Fig. 1(a). We assume that 101 moments are given (i.e., $M = 100$). Tables I–III show the average RMSE per each vertex for noise factor $\sigma_u = 1e-3$, $\sigma_u = 1e-4$, and $\sigma_u = 1e-5$, respectively. This variance represents the energy of a real Gaussian noise added to the real and the imaginary parts

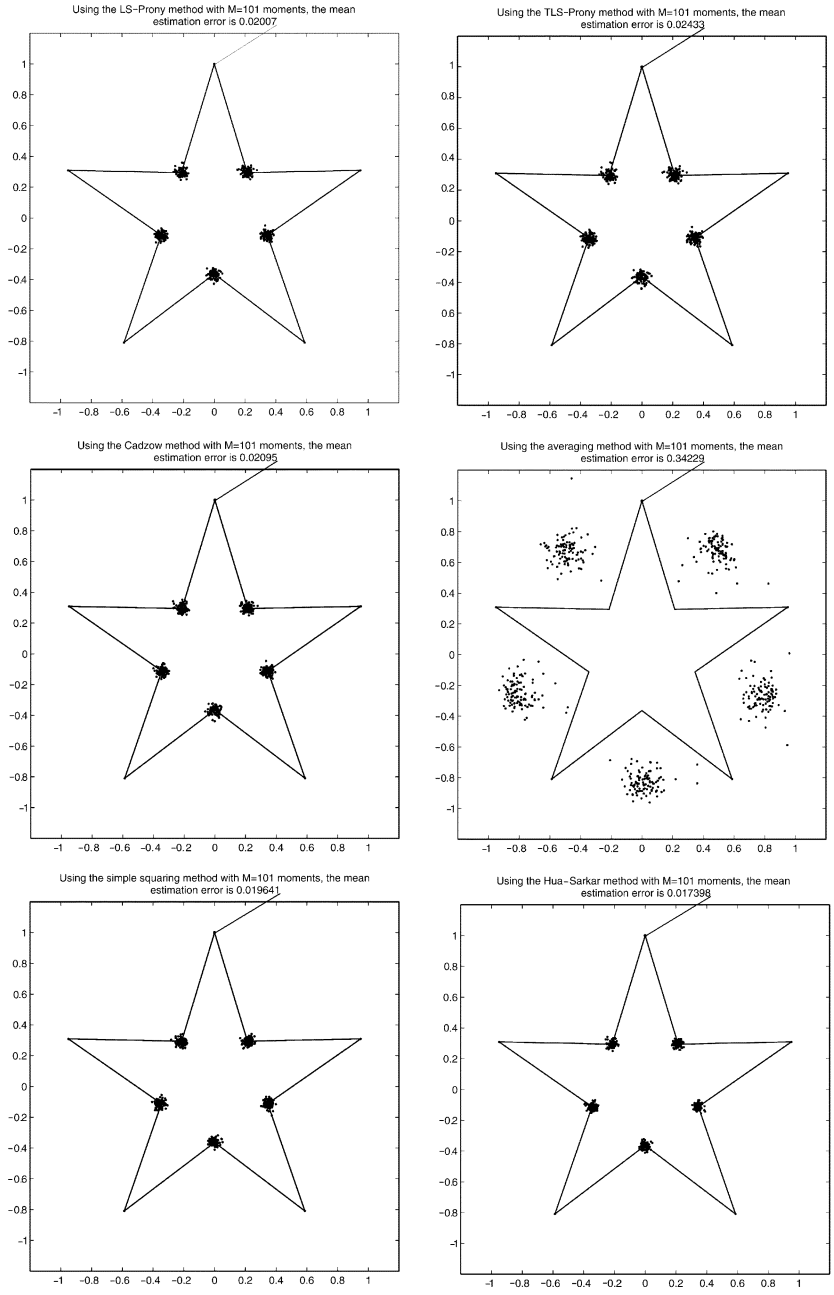


Fig. 2. Estimation results for the star shape with $\sigma_u = 1e-4$.

of the moment values. Table IV summarizes the overall error per each method and per each noise variance. Note that the smallest error achieved is highlighted in bold. Fig. 2 presents the location of the 100 estimated sets of vertices per each method for the case of $\sigma_u = 1e-4$.

Experiment 2: A similar experiment on an "E" shape shown in Fig. 1(b) and the estimation results are described in Tables V–VIII and in Fig. 3, following the same structure. This time, we assumed that 81 moments are given (i.e., $M = 80$). Fig. 3 corresponds to results obtained for $\sigma_u = 1e-3$, which is a higher noise compared with the one used in the creation of Fig. 2. This noise variance value was chosen in order to obtain larger error and, thus, a richer visual effect, as we found the results for this shape to be more robust.

In addition, the effect of changing the value of L in the GPOF algorithm is studied in Fig. 4 for this example. It can be seen that the error is relatively insensitive over a large range of L and noise variance values. Note that the left corner value corresponds to $L = N$, where we effectively implement the LS-Prony algorithm. The benefit of higher values of L is thus evident.

From these results, we can conclude the following.

- It is clear that the Pencil averaging method that we proposed based on intuition is very sensitive and does not show any promise. This result is very surprising, as for the exact moments, we know that each and every such square pencil among those averaged should have resulted in the proper solution. This goes to show that in passing from simple additive noise model to an eigenvalue perturbation,

TABLE V
E-SHAPE WITH NOISE FACTOR $\sigma_u = 1e-3$—RMSE PER VERTEX

Method	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8
LS-Prony	1.7e-3	3.1e-4	7.2e-2	2.1e-1	2.1e-1	7.2e-2	3.1e-4	1.7e-3
TLS-Prony	3.9e-4	9.5e-5	2.3e-2	5.2e-2	5.1e-2	2.4e-2	9.4e-5	3.8e-4
Cadzow	1.9e-4	3.8e-5	1.3e-2	2.5e-2	2.6e-2	1.3e-2	4.0e-5	1.9e-4
Pencil Ave.	7.0e-4	7.7e-4	2.8e-1	8.3e-1	8.0e-1	2.7e-1	8.3e-4	8.1e-4
Pencil Squaring	1.7e-3	3.2e-4	7.1e-2	2.1e-1	2.1e-1	7.2e-2	3.2e-4	1.7e-3
GPOF method	**7.6e-5**	**1.1e-5**	**3.8e-3**	**7.6e-3**	**8.4e-3**	**4.0e-3**	**1.0e-5**	**7.8e-5**

TABLE VI
E-SHAPE WITH NOISE FACTOR $\sigma_u = 1e-4$—RMSE PER VERTEX

Method	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8
LS-Prony	4.5e-5	1.1e-5	2.8e-3	6.1e-3	6.1e-3	2.8e-3	1.1e-5	4.3e-5
TLS-Prony	3.9e-5	9.5e-6	2.2e-3	5.1e-3	5.1e-3	2.2e-3	9.7e-6	3.8e-5
Cadzow	1.8e-5	3.3e-6	1.0e-3	2.2e-3	2.3e-3	1.1e-3	3.4e-6	1.8e-5
Pencil Ave.	1.1e-4	1.6e-4	2.4e-1	7.8e-1	7.8e-1	2.4e-1	1.8e-4	8.9e-5
Pencil Squaring	4.2e-5	1.0e-5	2.7e-3	6.0e-3	5.9e-3	2.7e-3	1.0e-5	4.3e-5
GPOF method	**7.7e-6**	**1.1e-6**	**4.0e-4**	**8.0e-4**	**8.1e-4**	**4.0e-4**	**1.0e-6**	**7.6e-6**

TABLE VII
E-SHAPE WITH NOISE FACTOR $\sigma_u = 1e-5$—RMSE PER VERTEX

Method	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8
LS-Prony	4.0e-6	8.7e-7	2.2e-4	5.1e-4	5.2e-4	2.2e-4	9.2e-7	4.1e-6
TLS-Prony	4.0e-6	8.8e-7	2.2e-4	5.1e-4	5.2e-4	2.2e-4	9.2e-7	4.0e-6
Cadzow	1.8e-6	3.2e-7	9.2e-5	2.1e-4	2.3e-4	1.0e-4	3.3e-7	1.9e-6
Pencil Ave.	1.1e-5	2.1e-5	2.2e-1	7.4e-1	7.5e-1	2.2e-1	2.9e-5	1.4e-5
Pencil Squaring	4.2e-6	9.6e-7	2.4e-4	5.5e-4	5.4e-4	2.3e-4	9.3e-7	3.9e-6
GPOF method	**7.7e-7**	**9.9e-8**	**3.7e-5**	**7.9e-5**	**8.7e-5**	**4.2e-5**	**9.8e-8**	**8.2e-7**

TABLE VIII
E-SHAPE—OVERALL RMSE PER METHOD

Noise variance	LS-Prony	TLS-Prony	Cadzow Alg.	Pencil Average	Pencil Squar.	GPOF method
1e-3	1.11e-1	2.82e-2	1.42e-2	4.30e-1	1.11e-1	**4.46e-3**
1e-4	3.35e-3	2.78e-3	1.24e-3	4.07e-1	3.27e-3	**4.51e-4**
1e-5	2.80e-4	2.81e-4	1.20e-4	3.87e-1	2.98e-4	**4.59e-5**

strong nonlinear and unstable effects take place. This also sheds some light on the sensitivity of the other algorithms to strong noise, as seen here.

- As expected, when noise variance is reduced, accuracy is improved. However, we see that most of the estimation methods do not degrade gracefully as the noise is increased. Rather, we see a strong nonlinear behavior according to which, for weak noise the algorithm behaves well, and for noise above some threshold the algorithm completely breaks down. We believe that analysis of the Cramér–Rao bound could explain this effect.
- Another effect related to the previous point is that the relative performance of the methods changes dramatically when the noise variance is changed. An ordering of the methods from the best to the worst is as follows: GPOF, Cadzow, TLS-Prony, and LS-Prony or the pencil-squaring method, which are roughly equivalent. In the TLS-Prony, we use the fact that all the moments are corrupted by noise and not just one column of the moments matrix. In the Cadzow algorithm, we exploit the Hankel-structure as well, and thus, we expect better performance than the TLS-Prony. In the GPOF method, we exploit the fact that

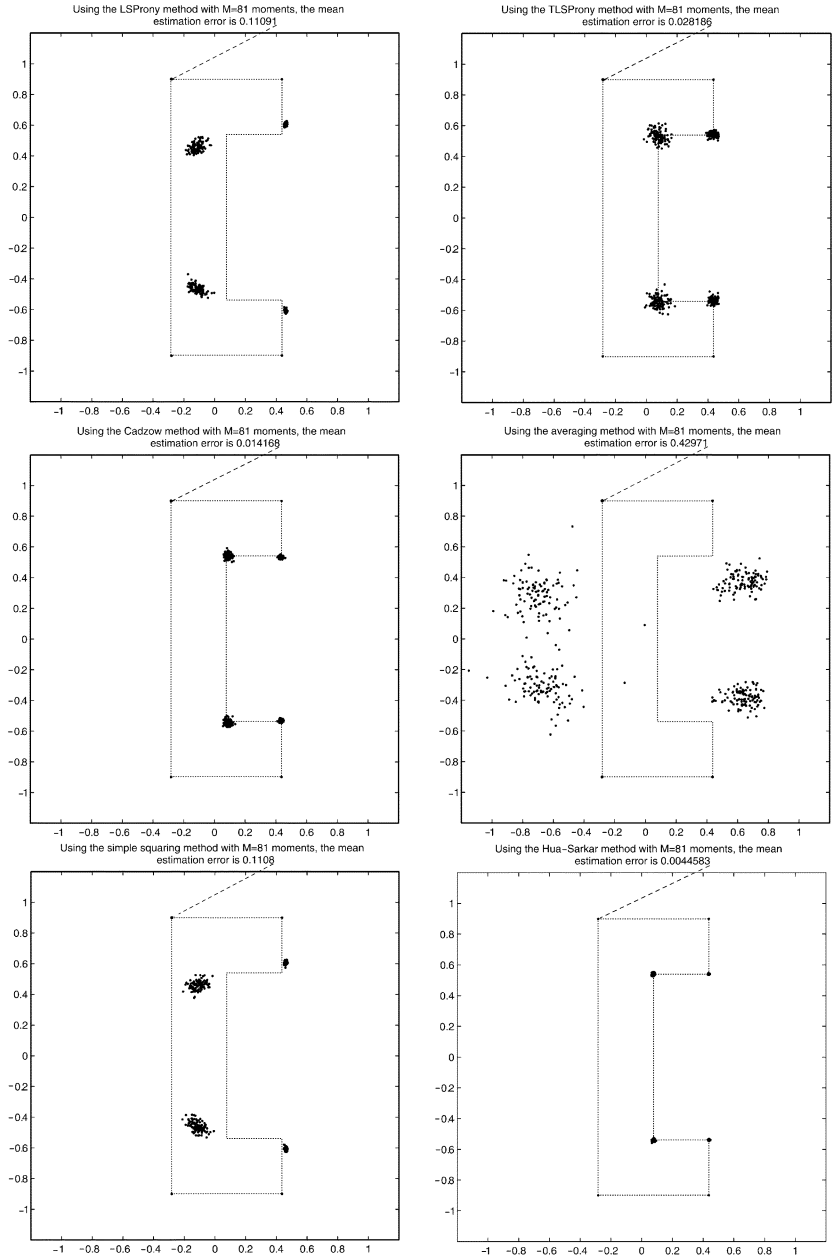


Fig. 3. Estimation results for the "E" shape with $\sigma_u = 1e-3$.

there is a large-dimensional subspace that should be orthonormal to the desired solution.

This ordering is indeed the one we find in our simulations, when the additive noise is weak enough. As the noise increases, however, the best performing algorithms (GPOF, Cadzow, and TLS-Prony) are more sensitive, whereas the LS-Prony and the pencil-squaring methods exhibit quite robust behavior. This effect of breakdown of the more accurate algorithms is more pronounced for the star-shape because it is symmetric. More on this effect is discussed next.

- In the star-shaped polygon, we consistently get better results with all methods for the outer vertices because they are convex. The overall error in the estimation is mostly driven by the error obtained for the inner vertices, which are the concave parts of the polygon. This can be explained by the wider angle that these vertices have and the fact that they form a concave part of the polygon (see [11] and [13] for more details on sensitivity in general and the geometric interpretation of it).
- Even though the overall behavior is similar in the two sets of experiments, it is clear that better accuracy is obtained for the "E" shape, where there is a sharper contrast between the best and the worst results. This is caused by the characteristics of the moment sequences involved. Since the star-shape is near-symmetrical, its moment-series contain many zeros. On the other hand, the "E" shape is less symmetric, and therefore, its moment series is more spread. As our estimation problem is essentially working with ratios between moments [see (8) or (17)], a moment series with zeros is expected to be much more vulnerable to errors.

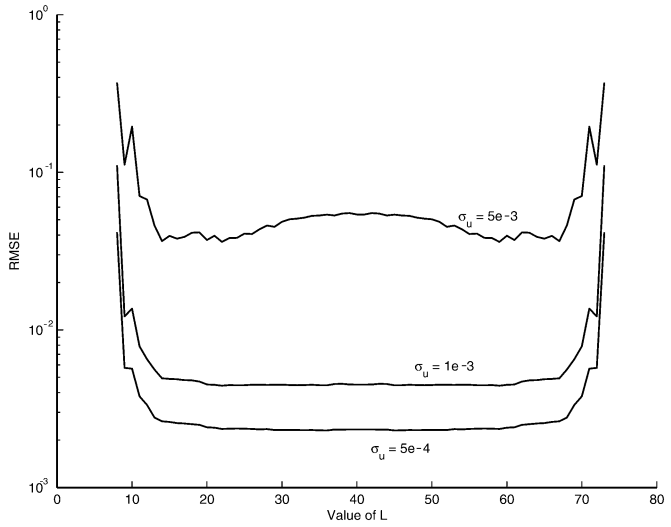


Fig. 4. Estimation results for the "E" shape with GPOF method and varying values of L.

- There is an interesting behavior common to all algorithms and found for very strong noise $(\sigma_u \gg 1)$. In this case, all the estimated eigenvalues tend to fall very close to the unit circle. The difference between the various algorithms is the breaking point when such a phenomenon starts to take place. In order to explain this effect, assume that we use the LS-Prony method. Using (8), we actually try to suggest an Nth-order linear predictor of a complex stochastic process sequence from its samples. The result is a random vector, and it is known (see [1]) that a polynomial with random coefficients lead to roots on the unit circle.

Experiment 3: We now present several results related to the improvements discussed in Section V. In this experiment, we return to the star shape and apply the GPOF method for obtaining an initialization solution. In this experiment, we assumed $\sigma_u = 1e-4$. We then update each vertex only once based on a local coordinate search, i.e., choosing the location corresponding to the nearest local minimum in a search window. Fig. 5 shows the behavior of the penalty function per each vertex while fixing all the other vertices. The location of the true, GPOF estimated (+ symbol), and improved vertices (dot symbol) are also overlayed. In this case, no regularization was used. The error obtained using the GPOF method is 0.0187, and after the update, it becomes 0.0074. It is interesting to see that the function we work with does not see fit to change the outer (convex) vertices; we know that these are estimated rather accurately.

Experiment 4: Table IX summarizes the results of the average error obtained over 20 runs using the star shape, applying the GPOF method for initialization, and applying 20 iterations of the coordinate descent algorithm. Each such iteration updates every vertex once, and therefore, we have 200 overall updates. As can be seen, the results are improved dramatically compared with the initialization.

Experiment 5: Again, using the "E" shape, assume $\sigma_u = 1e-3$, and we initialize using the LS-Prony algorithm. Similar to the process that created Figs. 5 and 6, the left side presents the ML function, as obtained by perturbing each vertex, assuming that all the others are fixed. We chose LS-Prony initialization and high noise variance in order to better see the errors and the achieved improvement. In this example, the error of the LS-Prony was found to be 0.106. After the improvement stage, we obtained an error of 0.062. Notice that we explore improvement in a finite size window and may obtain our optimal result on the boundary of this window, as indeed happens in this figure.

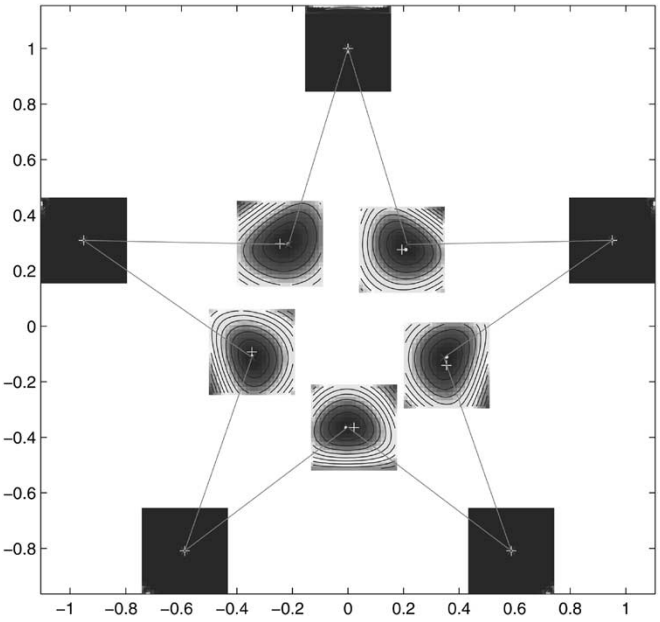


Fig. 5. Star shape estimation improvement—the penalty function as a function of each vertex separately.

TABLE IX
STAR-SHAPE—OVERALL RMSE FOR THE GPOF METHOD (USED AS INITIALIZATION) AND THE DIRECT ML APPROACH

Noise variance	GPOF method	Direct ML method
1e-3	3.70e-1	**3.19e-1**
1e-4	1.49e-2	**5.41e-3**
1e-5	1.72e-3	**2.12e-4**

Experiment 6: Table X summarizes the results obtained for the E-shape using the GPOF method as initialization and 20 iterations of the coordinate descent algorithm. Again, we averaged 20 experiments in order to see the aggregate effect of the direct ML algorithm. As can be seen, there is a consistent and marked improvement again.

Experiment 7: For the "E" shape, we add a regularization term promoting 90° angles, as proposed in Section V-B. This term is added to the function in (31) with a regularization coefficient found empirically to be 1000. The result as a function of each of the vertices separately is shown on the right side of Fig. 6, and indeed, the error reduced further to 0.041 using a simple and single coordinate descent update.

Experiment 8: The regularization impact on the results is also summarized in Table X, averaged again over 20 experiments using the same regularization function. For $\sigma_u = 1e-3$, we chose $\lambda = 1000$. As the noise change its energy, we change λ accordingly such that $\sigma \cdot \lambda$ is a constant. We see a consistent improvement due to the regularization, as expected.

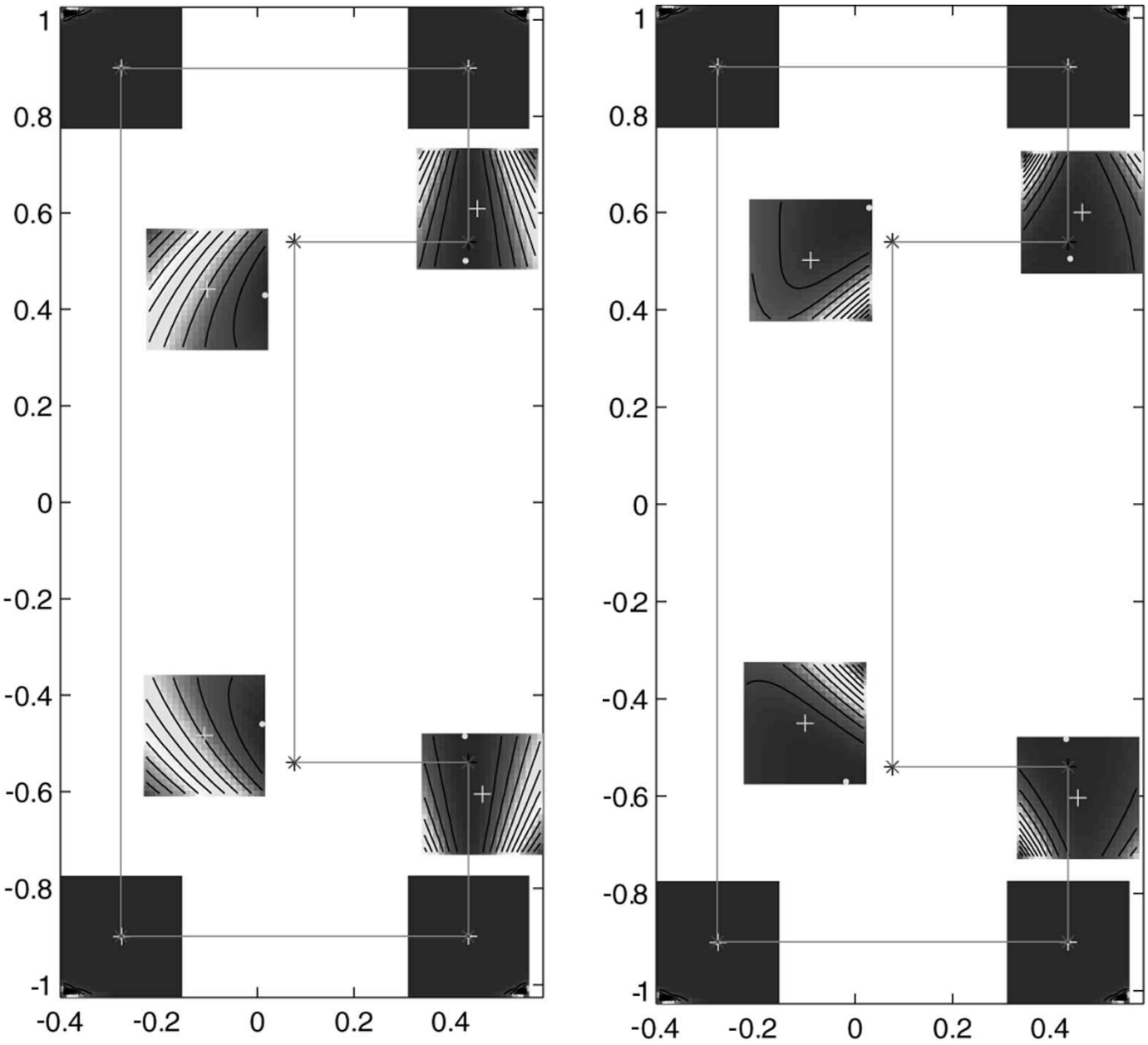


Fig. 6. "E" shape estimation improvement. (Left) Direct ML approach. (Right) Adding regularization.

TABLE X
E-SHAPE OVERALL RMSE FOR GPOF AS INITIALIZATION, THE DIRECT ML, AND THE MAP APPROACHES

Noise variance	GPOF method	Direct ML method	MAP method
1e-3	4.15e-3	2.16e-3	**1.64e-3**
1e-4	4.04e-4	3.13e-4	**2.85e-4**
1e-5	4.48e-5	1.23e-5	**1.13e-5**

VII. CONCLUDING REMARKS

This paper discusses the problem of reconstructing a planar polygon from its measured moments. When these moments are contaminated by additive noise, estimation procedures are required. Two families of such estimation algorithms are presented: the Prony- and the Pencil-based methods. For both families, we show that several algorithms emerge as candidate solvers. Simulations comparing these methods indicate that the GPOF method is indeed the leading method in accuracy in most cases.

We also find here that the least-squares Prony is generally exhibiting good and robust behavior when there is strong noise. This result is rather intriguing, as the literature dealing with this estimation problem typically tends to

rank the LS-Prony as one of the worst methods. Claims are also made on its relative weakness when compared with the pencil-based methods.

Finally, in this work, we proposed a method to improve the estimated vertices beyond the ability of the above-described methods, exploiting more features of the underlying problem. In this way, we showed that regularization can become part of the recovery process in a natural and effective way. We note that even though the applications from array processing and signal processing have a somewhat different formulation due to the lack of dependency of the coefficients on the vertices, we believe that the results presented here apply to these works as well. Further work is required in order to analyze these proposed extensions and improvements.

REFERENCES

[1] E. Bogomolny, O. Bohigas, and P. Leboeuf, "Distribution of roots of random polynomials," *Phys. Rev. Lett.*, vol. 68, no. 18, pp. 2726–2729, May 1992.

[2] Y. Bresler and A. Macovski, "Exact maximum likelihood parameter estimation of superimposed exponential signals in noise," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 1081–1089, Oct. 1986.

[3] J. A. Cadzow, "Signal enhancement: A composite property mapping algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 49–62, Jan. 1988.

[4] P. Davis, "Triangle formulas in the complex plane," *Math. Comput.*, vol. 18, pp. 569–577, 1964.

[5] ——, "Plane regions determined by complex moments," *J. Approx. Theory*, vol. 19, pp. 148–153, 1977.

[6] B. De Moor, "Structured total least squares and L_2 approximation problems," *Linear Algebra Applicat.*, vol. 188–189, pp. 163–205, July–Aug. 1993.

[7] S. Durocher, "Graph-theoretic and geometric algorithms associated with moment-based polygon reconstruction," M.S. thesis, Comput. Sci. Dept., Univ. British Columbia, Vancouver, BC, Canada, 1999.

[8] G. H. Golub and V. Pereyra, "The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate," *SIAM J. Numerical Anal.*, vol. 10, no. 2, pp. 413–432, 1973.

[9] G. H. Golub, M. Heath, and G. Wahba, "Generalized cross validation as a method for choosing a good ridge parameter," *Technometrics*, vol. 21, pp. 215–224, 1979.

[10] G. H. Golub and C. Van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: The Johns Hopkins Univ. Press, 1996.

[11] G. H. Golub, P. Milanfar, and J. Varah, "A stable numerical method for inverting shape from moments," *SIAM J. Sci. Comput.*, vol. 21, no. 4, pp. 1222–1243, Dec. 1999.

[12] G. H. Golub and V. Pereyra, "Separable nonlinear least squares and the variable projection method," *Inverse Problems*, vol. 19, no. 2, pp. R1–R26, Apr. 2003.

[13] B. Gustafsson, C. He, P. Milanfar, and M. Putinar, "Reconstructing planar domains from their moments," *Inverse Problems*, vol. 16, no. 4, pp. 1053–1070, Aug. 2000.

[14] F. B. Hildebrand, *Introduction to Numerical Analysis*, Second ed. New York: Dover, 1987.

[15] Y. Hua and T. K. Sarkar, "Matrix pencil method and its performance," in *Proc. IEEE ICASSP*, 1988, pp. 2476–2479.

[16] ——, "Generalized pencil-of-function method for extracting poles of an EM system from its transient response," *IEEE Trans. Antennas Propagat.*, vol. 37, pp. 229–234, Feb. 1989.

[17] ——, "Matrix pencil method for estimating parameters of exponentially damped/undamped sinusoids in noise," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 814–824, May 1990.

[18] ——, "On SVD for estimating generalized eigenvalues of singular matrix pencil in noise," *IEEE Trans. Signal Processing*, vol. 39, pp. 892–900, Apr. 1991.

[19] M. Kahn, M. S. Mackisack, M. R. Osborne, and G. K. Smyth, "On the consistency of Prony's method and related algorithms," *J. Comput. Graphical Statist.*, vol. 1, pp. 329–349, 1992.

[20] L. Kaufman and V. Pereyra, "A method for separable nonlinear least squares problems with separably nonlinear equality constraints," *SIAM J. Numer. Anal.*, vol. 15, pp. 12–20, 1978.

[21] P. Lemmerling, B. De Moor, and S. Van Huffel, "On the equivalence of constrained total least squares and structured total least squares," *IEEE Trans. Signal Processing*, vol. 44, pp. 2908–2911, Nov. 1996.

[22] P. Lemmerling and S. Van Huffel, "Analysis of the structured total least squares problem for Hankel/Toeplitz matrices," *Numerical Algorithms*, vol. 27, no. 1, pp. 89–114, 2001.

[23] P. Lemmerling, L. Vanhamme, S. Van Huffel, and B. De Moor, "IQML-like algorithms for solving structured total least squares problems: A unified view," *Signal Process.*, vol. 81, no. 9, pp. 1935–45, Sept. 2001.

[24] Y. Li, K. J. R. Liu, and J. Razavilar, "A parameter estimation scheme for damped sinusoidal signals based on low-rank Hankel approximation," *IEEE Trans. Signal Processing*, vol. 45, pp. 481–486, Feb. 1997.

[25] M. S. Mackisack, M. R. Osborne, and G. K. Smyth, "A modified Prony algorithm for estimating sinusoidal frequencies," *J. Statist. Comput. Simulation*, vol. 49, pp. 111–124, 1994.

[26] P. Milanfar, G. Verghese, W. Karl, and A. Willsky, "Reconstructing polygons from moments with connections to array processing," *IEEE Trans. Signal Processing*, vol. 43, pp. 432–443, Feb. 1995.

[27] M. R. Osborne, "Some special nonlinear least squares problems," *SIAM J. Numerical Anal.*, vol. 12, pp. 571–592, 1975.

[28] M. R. Osborne and G. K. Smyth, "A modified Prony algorithm for fitting functions defined by difference equations," *SIAM J. Sci. Statist. Comput.*, vol. 12, pp. 362–382, 1991.

[29] B. Porat, *A Course in Digital Signal Processing*. New York: Wiley, 1997.

[30] K. J. R. Liu, D. P. O'Leary, G. W. Stewart, and Y. J. Wu, "URV ESPRIT for tracking time-varying signals," *IEEE Trans. Signal Processing*, vol. 42, pp. 3441–3448, Dec. 1994.

[31] R. Roy and T. Kailath, "ESPRIT-estimation of signal parameters via rotational invariance techniques," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 984–995, July 1989.

[32] R. Roy, A. Paulraj, and T. Kailath, "ESPRIT: A subspace rotation approach to estimation of parameters of cissoids in noise," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-34, pp. 1340–1342, May 1986.

[33] L. L. Scharf, *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*. Reading, MA: Addison-Wesley, 1991.

[34] J. Razavilar, Y. Li, and K. J. R. Liu, "Structured low-rank matrix pencil for spectral estimation and system identication," *Signal Process.*, vol. 65, pp. 363–372, Mar. 1998.

[35] S. Van Huffel, H. Park, and J. B. Rosen, "Formulation and solution of structured total least norm problems for parameter estimation," *IEEE Trans. Signal Processing*, vol. 44, pp. 2464–2474, Oct. 1996.

Michael Elad received the B.Sc, M.Sc., and D.Sc. degrees from the Department of Electrical Engineering, the Technion–Israel Institute of Technology, Haifa, in 1986, 1988, and 1997, respectively.

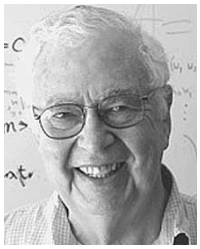
From 1988 to 1993, he served in the Israeli Air Force. From 1997 to 2000, he was with Hewlett-Packard Laboratories as an R&D engineer. From 1998 to 2000, he was also a research associate at the Technion, teaching courses in the Electrical Engineering Department. From 2000 to 2001, he headed the research division at Jigami Corporation, Israel. From 2001 to 2003, he was a research associate with the Computer Science Department (SCCM program), Stanford University, Stanford, CA. Since September 2003, he has been with the Department of Computer Science, the Technion, as an assistant professor. He works in the field of signal and image processing, specializing in particular on inverse problems, sparse representations and over-complete transforms, and image restoration algorithms.

Dr. Elad received the best lecturer award in 1999 and 2000.

Peyman Milanfar (SM'98) received the B.S. degree in electrical engineering/mathematics from the University of California, Berkeley, in 1988 and the S.M., E.E., and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1990, 1992, and 1993, respectively.

Until 1999, he was a Senior Research Engineer at SRI International, Menlo Park, CA. He is currently Associate Professor of electrical engineering at the University of California, Santa Cruz. He was a Consulting Assistant Professor of computer science at Stanford University, Stanford, CA, from 1998 to 2000 and a visiting Associate Professor there from June to December 2002. His technical interests are in the application of statistical signal processing methods to inverse problems in imaging.

Dr. Milanfar received a National Science Foundation CAREER award in 2000 and was an associate editor for the IEEE SIGNAL PROCESSING LETTERS from 1998 to 2001.



Gene H. Golub was born in Chicago, IL, in 1932. He received the Ph.D. degree from the University of Illinois, Urbana, in 1959.

He held an NSF Fellowship at the University of Cambridge, Cambridge, U.K. He joined the faculty of Stanford University, Stanford, CA, in 1962, where he is currently the Fletcher Jones Professor of Computer Science and the Director of the Program in Scientific Computing and Computational Mathematics. He served as Chairman of the Computer Science Department from 1981 until 1984. He is noted for his work in the use of numerical methods in linear algebra for solving scientific and engineering problems. This work resulted in a book entitled *Matrix Computations* (Baltimore, MD: Johns Hopkins University Press, 1996) co-authored with C. Van Loan. He served as President of the Society of Industrial and Applied Mathematics from 1985 to 1987 and is the Founding Editor of the *SIAM Journal of Scientific and Statistical Computing* and the *SIAM Journal of Matrix Analysis and Applications*. He is the originator of na-net.

Dr. Golub holds several honorary degrees. He is a member of the National Academy of Engineering, the National Academy of Sciences, and the American Academy of Sciences.