TWO-DIMENSIONAL PRONY MODELING AND PARAMETER ESTIMATION *

Joseph J. Sacchini *William M. Steedly* *Randolph L. Moses*

Department of Electrical Engineering
The Ohio State University
Columbus, Ohio 43210, USA

ABSTRACT

A new method for estimating two-dimensional (2-D) poles and amplitude coefficients in a Prony model is presented. This method involves two parts, each utilizing a 1-D singular value decomposition-based technique, and is capable of locating frequencies anywhere in the 2-D frequency plane. Simulations are shown which demonstrate the performance of the algorithm.

1. INTRODUCTION

For many years the problem of two-dimensional (2-D) frequency and amplitude coefficient estimation from a 2-D data set has been investigated. This problem has applications in sonar, radar, geophysics, radio astronomy, radio communications, and medical imaging [1]. Many techniques have been applied to the problem such as Fourier-based methods, data extension, maximum likelihood method (MLM), maximum entropy method (MEM), autoregressive (AR) models, and linear prediction (LP) models [2].

Prony's method coupled with total least squares (TLS) techniques in one-dimension (1-D) has been used successfully to estimate frequencies in the presence of noise [4]. In this paper, 2-D frequencies and amplitude coefficients are estimated by a two-step method using a 1-D TLS-based Prony model and estimation technique in each step. This method is capable of locating frequencies anywhere in the 2-D plane.

A related method, developed by Hua [3], also estimates 2-D frequencies. In Hua's method, two estimation steps are performed to separately estimate the x-components and y-components of the 2-D frequencies. Then, a matching step is performed to find the correct x and y-component frequency pairings. The method we present is similar to Hua's in some respects, but different in others. We first estimate the x-components of the frequencies. We then use the amplitude coefficients corresponding to each x-component frequency to estimate a set of y-components. In this way we avoid the requirement of a matching step. Our algorithm is computationally less expensive than Hua's method, and more amenable to parallel implementation. We also present a second algorithm in which the first algorithm is used twice, first to estimate x then y-components and then y then x-components. The second algorithm gives more accurate parameter estimates than the first algorithm. This second algorithm does require matching, but the computational load is still significantly smaller than the one in [3].

2. DATA MODEL AND ESTIMATION ALGORITHMS

2.1. Data Model

Assume we are given noisy 2-D data which has the form

$$d'(m,n) = d(m,n) + w(m,n), \tag{1}$$

where $m = 0, 1, \dots M-1$ and $n = 0, 1, \dots N-1$ and $w(m,n)$ is 2-D noise sequence. We will refer to the first index of $d(m,n)$ as the x-component, and the second index as the y-component.

The noiseless data is assumed to fit the damped exponential model

$$d(m,n) = \sum_{k=1}^{K} \sum_{l=1}^{L_k} a_{k,l} p_{x_k}^m p_{y_{k,l}}^n, \tag{2}$$

where

$$\begin{aligned} p_{x_k} &= k\text{th } x\text{-pole, } x\text{-component of 2-D exponential} \\ p_{y_{k,l}} &= k,l\text{th } y\text{-pole, } y\text{-component of 2-D exponential} \\ a_{k,l} &= k,l\text{th amplitude coefficient} \\ L_k &= \#\text{ of } y\text{-poles corresponding to } k\text{th } x\text{-pole.} \end{aligned} \tag{3}$$

Given noisy data $d'(m,n)$, we wish to estimate the parameters in Equation 2. Below we present two TLS-based algorithms for estimating these parameters, Algorithm One and Algorithm Two.

2.2. Algorithm One

Algorithm One consists of four steps detailed below.

Step 1: Estimation of the x-poles.

The first step in the parameter estimation problem is to estimate the x-poles. We now define a matrix composed of the noisy data as

$$D' = \begin{bmatrix} d'(0,0) & d'(1,0) & \cdots & d'(M-1,0) \\ d'(0,1) & d'(1,1) & \cdots & d'(M-1,1) \\ \vdots & \vdots & & \vdots \\ d'(0,N-1) & d'(1,N-1) & \cdots & d'(M-1,N-1) \end{bmatrix}. \tag{4}$$

Each row of D' can be used to provide an estimate of the x-poles. However, all of the rows of D' will be used simultaneously in the estimation of the x-poles. A total least squares (TLS) backward linear prediction approach similar to [4] is used. The backward linear prediction equations are

$$\begin{bmatrix} D'^T(0) \\ D'^T(1) \\ \vdots \\ D'^T(N-1) \end{bmatrix} \begin{bmatrix} 1 \\ b_1 \\ \vdots \\ b_Q \end{bmatrix} \approx 0 \tag{5}$$

*THIS RESEARCH WAS SUPPORTED IN PART BY THE AIR FORCE OFFICE OF SCIENTIFIC RESEARCH, BOLLING AFB, DC, AND IN PART BY THE AVIONICS DIVISION, WRIGHT LABORATORIES, WRIGHT PATTERSON AFB, OH.

0-7803-0532-9/92 $3.00 © 1992 IEEE

or $S \begin{bmatrix} 1 \\ b \end{bmatrix} \approx 0$, where each $D'(n)$ is a Toeplitz matrix given by

$$D'(n) = \begin{bmatrix} d'(0,n) & d'(1,n) & \cdots & d'(Q,n) \\ d'(1,n) & d'(2,n) & \cdots & d'(Q+1,n) \\ \vdots & \vdots & & \vdots \\ d'(M-Q-1,n) & d'(M-Q,n) & \cdots & d'(M-1,n) \end{bmatrix}$$

and where Q is the order of prediction, and b is the coefficient vector of the polynomial $B_Q(z)$ given by

$$B_Q(z) = 1 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_Q z^{-Q}. \qquad (6)$$

Ideally, Q can be any integer greater than or equal to the model order K; in practice, choosing $Q > K$ results in more accurate parameter estimates [5]. Note that all of the rows of D' are used simultaneously to estimate a single set of prediction coefficients (and therefore, a single set of x-poles).

Equation 5 is used to solve for $\widehat{b}$ in a total least squares sense to arrive at a minimum norm (TLS) estimate of $\widehat{b}$, where the $Q+1-K$ smallest singular values of S are truncated to arrive at a noise cleaned estimate $\widehat{S}$ (see [4] for details).

The estimated x-poles are found by

$$\widehat{p}_{x_q} = \left(\text{zero}_q\left(\widehat{B_Q(z)}\right)\right)^{-1}, \qquad q = 1, 2, \ldots, Q. \qquad (7)$$

Of these Q poles, only the K x-poles which have the largest energy are retained.

Step 2: Estimation of the x-amplitude Coefficients.

Define the x-amplitude coefficients as

$$c_{q,n} = \sum\nolimits_{l=1}^{L_k} a_{q,l} p_{y_{q,l}}^n, \qquad q = 1, 2, \ldots, Q. \qquad (8)$$

Then from Equation 2 we have

$$d(m,n) = \sum\nolimits_{q=1}^{Q} c_{q,n} p_{x_q}^m. \qquad (9)$$

Note that the equations in 9 are uncoupled for different values of n. Thus, each row of D' will give an x-amplitude coefficient estimate for each x-pole. These estimates serve as the inputs to the second Prony model and determine the y-poles. Also note that the y-pole model orders, $\{L_k\}_{k=1}^{K}$, may be different for each of the K x-poles.

Equation 9 is used to solve for the $c_{q,n}$s as follows

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ p_{x_1} & p_{x_2} & \cdots & p_{x_Q} \\ \vdots & \vdots & & \vdots \\ p_{x_1}^{M-1} & p_{x_2}^{M-1} & \cdots & p_{x_Q}^{M-1} \end{bmatrix} \begin{bmatrix} c_{1,0} & c_{1,2} & \cdots & c_{1,N-1} \\ c_{2,0} & c_{2,2} & \cdots & c_{2,N-1} \\ \vdots & \vdots & & \vdots \\ c_{Q,0} & c_{Q,1} & \cdots & c_{Q,N-1} \end{bmatrix} = D'^T. \qquad (10)$$

The x-amplitude coefficients are found from a QR decomposition based least squares solution to Equation 10 using the x-pole estimates.

Step 3: Estimation of the y-poles.

The x-amplitude coefficients can now be used to solve for the y-poles. For each of the K high energy x-poles, the backward linear prediction equations for the model given by Equation 8 become

$$\begin{bmatrix} c_{k,0} & c_{k,1} & \cdots & c_{k,R_k} \\ c_{k,1} & c_{k,2} & \cdots & c_{k,R_k+1} \\ \vdots & \vdots & & \vdots \\ c_{k,N-R_k-1} & c_{k,N-R_k} & \cdots & c_{k,N-1} \end{bmatrix} \begin{bmatrix} 1 \\ b_1^k \\ \vdots \\ b_{R_k}^k \end{bmatrix} \approx 0, \qquad (11)$$

where R_k is the order of prediction for the y-poles, and b^k is the coefficient vector of the polynomial $B_{R_k}^k(z)$.

Equation 11 is used to solve $\widehat{b^k}$ in a total least squares sense to arrive at a minimum norm (TLS) estimate of $\widehat{b^k}$, where the $R_k + 1 - L_k$ singular values of S are truncated.

The y-pole estimates are thus given by inverses of the zeros of $\widehat{B^k}(z)$ as in Equation 7. This procedure is carried out K times to estimate the y-poles corresponding to each of the K x-poles.

Step 4: Estimation of the Amplitude Coefficients.

Using Equation 8, we can write

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ p_{y_{k,1}} & p_{y_{k,2}} & \cdots & p_{y_{k,R_k}} \\ \vdots & \vdots & & \vdots \\ p_{y_{k,1}}^{N-1} & p_{y_{k,2}}^{N-1} & \cdots & p_{y_{k,R_k}}^{N-1} \end{bmatrix} \begin{bmatrix} a_{k,1} \\ a_{k,2} \\ \vdots \\ a_{k,R_k} \end{bmatrix} = \begin{bmatrix} c_{k,0} \\ c_{k,1} \\ \vdots \\ c_{k,N-1} \end{bmatrix}. \qquad (12)$$

The amplitude coefficients are found from a QR decomposition based least squares solution to Equation 12 using the y-pole estimates along the x-amplitude coefficients. As before, only the L_k y-poles which have the largest energy are retained. This is done by computing the R_k y-mode energies for each of the kth x-poles and retaining those L_k poles whose corresponding energies are highest.

2.3. Algorithm Two

Algorithm Two utilizes the first three steps of Algorithm One twice and then a matching step and final amplitude coefficient calculation step. The original data, $d'(m,n)$, is the input to Algorithm One. The first three steps are carried out yielding x and y-pole estimates, $\{p_{x_k}\}_{k=1}^{K}$ and $\{p_{y_{k,l}}\}_{l=1}^{L_k}$. Next, the data is transposed (*i.e.* $d''(n,m) = d'(m,n)$), and Algorithm One is applied to $d''(n,m)$ to arrive at a second set of poles, $\{p_{x_{k^t}}^t\}_{k^t=1}^{K^t}$ and $\{p_{y_{k^t,l^t}}^t\}_{l^t=1}^{L_{k^t}^t}$. Note that the model orders K and L_k are related to K^t and $L_{k^t}^t$, depending on the structure of a particular model, and are in general different.

The two sets of estimates are combined, and more accurate part of the estimates from each set is retained. The more accurate part of each estimate is the set of poles which were estimated first. A matching algorithm is used to combine both sets of pole estimates yielding a single set of pole estimates.

The matching is performed using the following metric

$$\Delta\left((p_{x_k}, p_{y_{k,l}}), (p_{x_{k^t}}^t, p_{y_{k^t,l^t}}^t)\right) = \sqrt{|p_{x_k} - p_{y_{k^t,l^t}}^t|^2 + |p_{y_{k,l}} - p_{x_{k^t}}^t|^2} \qquad (13)$$

for the distance between 2-D exponential modes $(p_{x_k}, p_{y_{k,l}})$ and $(p_{x_{k^t}}^t, p_{y_{k^t,l^t}}^t)$, estimated from the two estimates in Step 1. These distances are calculated for all of the possible pairs. Then the closest match is made and the respective pole pairs and distances are eliminated from consideration. The next-to-closest match is then made in the same fashion and so on, until there are no modes remaining from one of the two sets of pole pairs parts (any leftover pole pairs are discarded). Note that the x-poles, p_{x_k} and $p_{x_{k^t}}^t$, from each of the two estimations are retained and the y-poles are discarded as discussed above. Thus, the y-poles are only necessary for the pairing performed in the matching step. Note that the $p_{x_{k^t}}^t$ take on the role of y-poles in the original model. The final set of matched pole pairs for Algorithm Two are thus designated $\{\rho_{x_\gamma}, \rho_{y_\gamma}\}_{\gamma=1}^{\Gamma}$, where the ρ's are given by the paired p_x's and p_x^t's and $\Gamma = \min\left\{\sum_{k=1}^{K} L_k, \sum_{k^t=1}^{K^t} L_{k^t}^k\right\}$.

Using this definition, the model in Equation 2 can be expressed as

$$d(m,n) = \sum_{\gamma=1}^{\Gamma} \alpha_\gamma \rho_{x_\gamma}^m \rho_{y_\gamma}^n. \tag{14}$$

Equation 14 is used to solve for amplitude coefficients, $\{\alpha_\gamma\}_{\gamma=1}^{\Gamma}$, as follows

$$\begin{bmatrix} P(0) \\ P(1) \\ \vdots \\ P(M-1) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_\Gamma \end{bmatrix} = \begin{bmatrix} d'(0) \\ d'(1) \\ \vdots \\ d'(M-1) \end{bmatrix}, \tag{15}$$

where each $P(m)$ and $d'(m)$ are given by

$$P(m) = \begin{bmatrix} \rho_{x_1}^m & \rho_{x_2}^m & \cdots & \rho_{x_\Gamma}^m \\ \rho_{x_1}^m \rho_{y_1} & \rho_{x_2}^m \rho_{y_2} & \cdots & \rho_{x_\Gamma}^m \rho_{y_\Gamma} \\ \vdots & \vdots & & \vdots \\ \rho_{x_1}^m \rho_{y_1}^{N-1} & \rho_{x_2}^m \rho_{y_2}^{N-1} & \cdots & \rho_{x_\Gamma}^m \rho_{y_\Gamma}^{N-1} \end{bmatrix} \tag{16}$$

and

$$d'(m) = \begin{bmatrix} d'^T(m,0) & d'^T(m,1) & \cdots & d'^T(m,N-1) \end{bmatrix}^T. \tag{17}$$

The amplitude coefficients are found using the pole estimates from a QR decomposition based least squares solution to Equation 15.

2.4. Implementation Issues

In this section we present operation counts for the four steps of Algorithm One and for the eight steps of Algorithm Two. These operation counts are given for the case when the data is real. For complex data considered in the examples below the counts were observed to be about a factor of 2 to 3 larger for the SVDs and about 4 times larger for the QR decompositions.

2.4.1. Operation Count for Algorithm One

To obtain the operation counts for the four steps of Algorithm One, we need counts for the SVD computations and the computation of the QR decompositions used for the least squares solutions.

The approximate floating point operation (flop) counts for Steps 1–4 where the given data is real are given by [6]

$$\begin{aligned} \mathrm{fc}_{\mathrm{One}}^1 &\approx 4N(M-Q)(Q+1)^2 + 8(Q+1)^3 \\ \mathrm{fc}_{\mathrm{One}}^2 &\approx 2MQ^2 - \frac{2}{3}Q^3 \\ \mathrm{fc}_{\mathrm{One}}^3 &\approx \sum_{k=1}^{K} \left(4(N-R_k)(R_k+1)^2 + 8(R_k+1)^3\right) \\ \mathrm{fc}_{\mathrm{One}}^4 &\approx \sum_{k=1}^{K} \left(2NR_k^2 - \frac{2}{3}R_k^3\right). \end{aligned} \tag{18}$$

To achieve near optimal performance (with respect to the CRB), the model order used for Steps 1 and 3 of Algorithm One should be integers near $Q = \frac{M}{3}$ and $R_k = \frac{N}{3}$ [7]. Using these substitutions and further approximations we arrive the following estimate for the total flop count

$$\mathrm{fc}_{\mathrm{One}} \approx \frac{1}{3}M^3N + \frac{1}{2}KN^3. \tag{19}$$

Note that the K estimations of Steps 3 and 4 are independent of each other and can thus be done in parallel.

2.4.2. Operation Count for Algorithm Two

Since Algorithm Two first utilizes the first three steps Algorithm One we obtain $\mathrm{fc}_{Two}^1 = \mathrm{fc}_{One}^1$, $\mathrm{fc}_{Two}^2 = \mathrm{fc}_{One}^2$, and $\mathrm{fc}_{Two}^3 = \mathrm{fc}_{One}^3$ for the flop counts of those steps. For Steps 4–6, the roles of M and N reversed, and Q^t and $R_{k^t}^t$ are used instead of Q and R_k. We thus obtain the following for their flop counts

$$\begin{aligned} \mathrm{fc}_{\mathrm{Two}}^4 &\approx 4M(N-Q^t)(Q^t+1)^2 + 8\left(Q^t+1\right)^3 \\ \mathrm{fc}_{\mathrm{Two}}^5 &\approx 2NQ^{t2} - \frac{2}{3}Q^{t3} \\ \mathrm{fc}_{\mathrm{Two}}^6 &\approx \sum_{k^t=1}^{K^t} \left(4\left(M - R_{k^t}^t\right)\left(R_{k^t}^t + 1\right)^2 + 8\left(R_{k^t}^t + 1\right)^3\right). \end{aligned}$$

The flop count for Step 7 is negligible compared to the other steps. For the QR decomposition in Step 8 the approximate flop count is given by

$$\mathrm{fc}_{\mathrm{Two}}^8 \approx 2MN\Gamma^2 - \frac{2}{3}\Gamma^3. \tag{20}$$

Again, to achieve near optimal performance (with respect to the CRB), the model order used for Steps 1, 3, 4, and 6 of Algorithm Two should be integers near $Q = R_{k^t}^t = \frac{M}{3}$, $R_k = Q^t = \frac{N}{3}$ [7]. Using these substitutions and further approximations we arrive the following estimate for the total flop count

$$\mathrm{fc}_{\mathrm{Two}} \approx \frac{1}{3}M^3N + \frac{1}{2}KN^3 + \frac{1}{3}N^3M + \frac{1}{2}K^tM^3 + 2MN\Gamma^2 - \frac{2}{3}\Gamma^3.$$

Note that the K estimations of Step 3 are independent and the K^t estimations of Step 6 are independent and can thus be done in parallel.

3. SIMULATIONS

Below we present numerical simulations to assess model validity and noise effects. The example considers the estimation of three 2-D undamped exponentials; this example was also considered in [3].

In this example we compare the variances of frequency estimates to their CRBs at various signal to noise ratios SNRs for the three 2-D frequency scenario presented in [3] using Algorithm Two. Data was generated using the model in Equation 2 for $M = N = 20$ and

$$\begin{aligned} \begin{bmatrix} p_{x_1} & p_{y_{1,1}} & a_{1,1} \end{bmatrix} &= \begin{bmatrix} e^{j2\pi 0.24} & e^{j2\pi 0.24} & 1 \end{bmatrix} \\ \begin{bmatrix} p_{x_1} & p_{y_{1,2}} & a_{1,2} \end{bmatrix} &= \begin{bmatrix} e^{j2\pi 0.24} & e^{j2\pi 0.26} & 1 \end{bmatrix} \\ \begin{bmatrix} p_{x_2} & p_{y_{2,1}} & a_{2,1} \end{bmatrix} &= \begin{bmatrix} e^{j2\pi 0.26} & e^{j2\pi 0.24} & 1 \end{bmatrix}. \end{aligned}$$

We can see that from the angles of the above poles the corresponding frequencies are 0.24 and 0.26. For the purposes of identification we will label the (x-pole,y-pole) pairs above with frequencies (0.24,0.24), (0.24,0.26), and (0.26,0.24) as 2-D frequencies f_1, f_2, and f_3, respectively. Note that these frequencies are spaced at four-tenths of a Fourier bin in both directions (1 Fourier bin$= \frac{1}{20} = 0.05$).

One hundred different noise realizations were run for each integer SNR between 0 and 50dB utilizing Algorithm Two. The SNR is defined as the total signal power divided by the total noise power. Figures 1 and 2 show the simulation results for the x-pole frequencies and y-pole frequencies, respectively. Specifically these figures show the estimated frequency variances for the various SNRs (they are given by the dashed lines as indicated). The corresponding CRBs are given by the solid lines; the CRBs of the model were found using the expressions in [8]. The algorithm parameters were set at $Q = 8, K = 2$, $R_1 = R_2 = 8$, $L_1 = L_2 = 2$ for this example. Note that we have used $Q \approx \frac{M}{3}$ and $R_k \approx \frac{N}{3}$ because these values give maximum parameter accuracy in the SVD estimation step [7].

From Figures 1 and 2 we can see that the threshold SNR is about 15dB. For SNRs above 15dB, the variances are within 4dB of their respective CRBs; below 15dB the algorithm fails to reliably resolve the frequencies. The simulation variance lines even cross the CRB due to the fact that the poles are no longer resolved and thus the estimates cannot be used in the variance calculations. The results in [3] are for an SNR per pole of 10dB, which corresponds to a total SNR of 17.7dB used here. In comparison, the estimation scheme in [3] gives variances which are about 3 or 4dB better for the x-pole frequencies of f_1 and f_3 and the y-pole frequencies of f_1 and f_2 and variances which are about 8dB better for the x-pole frequency of f_2 and the y-pole frequency of f_3.

We next consider the SVD operation counts for this example. The SVD steps are major computational parts. Each noise realization required SVDs of two 240×9 matrices and four 9×11 matrices where only the singular values and right singular vectors were computed. For the same example, the algorithm in [3] required SVDs of two 49×169 matrices for each noise realization where the singular values and left singular vectors are computed. Looking at expressions in [6] for SVD computations we can see that Algorithm Two requires significantly fewer flops than the algorithm in [3] for the SVDs. We note that the SVDs for both algorithms can be performed with fewer computations by computing the eigendecomposition of smaller square matrices. Using this idea the total number of computations are reduced for both methods and there is still significant savings with Algorithm Two over the algorithm in [3].

4. CONCLUSIONS

We have presented a new method for estimating 2-D poles and amplitude coefficients. This method utilizes a 1-D TLS-based Prony model and estimation technique. This process has computational advantages over methods which have independent steps such as the one in [3] since the second step involves several smaller estimations rather than one estimation as large as the first. This procedure also has the advantage not requiring a pairing of x and y-poles.

Simulations demonstrate the algorithms ability to estimate 2-D poles and amplitude coefficients in the presence of noise reasonably well. The estimates in [3] are slightly more accurate than the estimates for this technique, but this technique offers significant computational savings over [3]. For Algorithm One, the y-pole estimation is less accurate than the x-pole estimation due to a propagation of error in the x-pole estimation. If more accuracy is required in both the x and y-pole estimates then Algorithm Two can be implemented which still has significant computational savings over [3].

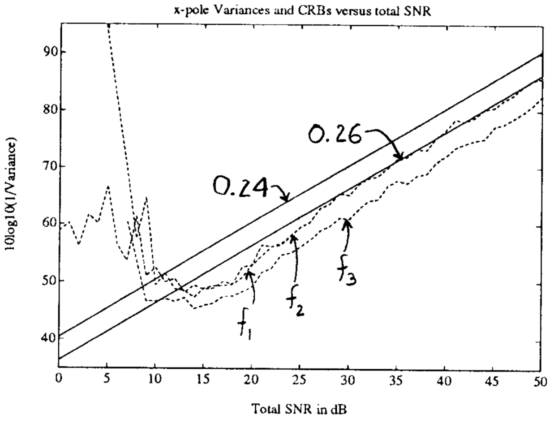

Figure 1: $10\log_{10}(1/\text{Variance})$ versus total SNR in dB for x-pole frequencies for Algorithm Two.

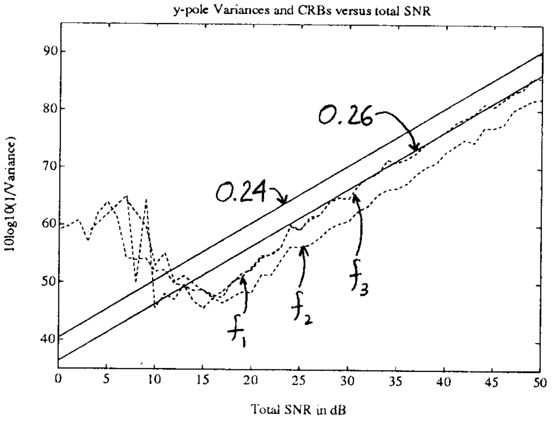

Figure 2: $10\log_{10}(1/\text{Variance})$ versus total SNR in dB for y-pole frequencies for Algorithm Two.

REFERENCES

[1] R. M. Lewitt, "Reconstruction algorithms: Transform methods," *Proc. IEEE*, vol. 71, pp. 390–408, Mar. 1983.

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

[3] Y. Hua, "Estimating two-dimensional frequencies by matrix enhancement and matrix pencil," in *Proc. ICASSP*, (Toronto), pp. 3073–3076, May 14-17 1991.

[4] M. A. Rahman and K.-B. Yu, "Total least squares approach for frequency estimation using linear prediction," *IEEE Trans. ASSP*, vol. 35, pp. 1440–1454, Oct. 1987.

[5] P. Stoica, T. Söderström, and F. Ti, "Asymptotic properties of the high-order Yule-Walker estimates of sinusoidal frequencies," *IEEE Trans. ASSP*, vol. 37, pp. 1721–1734, Nov. 1989.

[6] G. H. Golub and C. F. V. Loan, *Matrix Computations – second edition.* Baltimore: The Johns Hopkins University Press, 1989.

[7] Y. Hua and T. K. Sarkar, "A perturbation property of the TLS-LP method," *IEEE Trans. ASSP*, vol. 38, pp. 2004–2005, Nov. 1990.

[8] W. M. Steedly and R. L. Moses, "The Cramér-Rao bound for pole and amplitude estimates of damped exponential signals in noise," in *Proc. ICASSP*, (Toronto), pp. 3569–3572, May 14-17, 1991.