Mathematics Notes

Note 59

November 1978

An Improved Prony Algorithm for Exponential Analysis

Harold J. Price
Mission Research Corporation
Albuquerque, New Mexico 87108

Abstract

An improved Prony algorithm is described which produces a lower sum of squares of residuals than the usual Prony Algorithm. Indeed, the new algorithm produces the lowest sum of squares of residuals possible. A simple numerical example is used to compare the new and old algorithms. Within the context of the new algorithm, a technique to constrain poles is also described.

Acknowledgement

The author wishes to acknowledge several illuminating discussions with V. van Lint of the San Diego Mission Research Corporation office.

CONTENTS

1. INTRODUCTION

The decomposition of a function into a sum of decaying sinusoids; that is, exponential analysis, can be a useful tool for the analysis of measured data, and for the rational-function approximation to frequency-domain data. The most popular algorithm for exponential analysis is a modification to the original algorithm established by Prony in 1795 [1], [2]. It is not the purpose here to justify exponential analysis; rather, we use this note to document several improvements to the modified Prony algorithm which results in better "fits" to data.

It is always possible to exactly fit 2N, finite, equally-spaced data points with N decaying exponentials; indeed, the original Prony algorithm will do this. Modification to the original algorithm is necessary where a finite sum of exponentials is desired and the number of terms is less than required to exactly fit the data. In this case, one looks for a "best" fit.

Other modifications are necessary to force (constrain) a subset of the poles to have specific values. For example, if it is known that certain poles are present in the data, then these poles can be forced to occur.

In order to illustrate the advantages of the new algorithm, we first describe the usual Prony algorithm and then proceed to derive the new algorithm. We also include a method to constrain poles.

2. CONVENTIONAL PRONY ALGORITHM

Like Dudley [2], we view the problem of exponential analysis as one of system identification. Accordingly, it is assumed that the system model is defined by a difference equation which has a polynomial as

solution. The variables in the polynomial are related to the desired exponential coefficients.

The actual data, when substituted into the model, does not in general satisfy the difference equation, but results in an error or residual. The coefficients of the difference equation are found by minimizing the sums of squares of these residuals.

Let the model difference equation have N coefficients, resulting in an $(N-1)^{th}$ order characteristic equation and N-1 Prony poles. When the data are substituted in the difference equation, the following equation results

$$D\vec{\alpha} = \vec{\delta} \quad ; \tag{1}$$

where D is an MxN matrix of equally-spaced data (in general, M>N),

$\vec{\alpha}$ is an Nx1 vector of coefficients,

$\vec{\delta}$ is an Nx1 vector of residuals.

We desire to find the solution $\vec{\alpha}$ which minimizes the sums of squares of the residuals; i.e., minimize the inner product $(\vec{\delta}, \vec{\delta})$.

From equation 1, this inner product is

$$\begin{aligned} (\vec{\delta}, \vec{\delta}) &= (D\vec{\alpha})^T (D\vec{\alpha}) \\ &= \vec{\alpha}^T D^T D\vec{\delta} \\ &= \vec{\alpha}^T A\vec{\alpha} \end{aligned} \tag{2}$$

Here A is the NxN matrix of the resulting quadratic form.

Obviously, a trivial solution to equation 2 is $\vec{\alpha} = \vec{0}$. In order to obtain a non-trivial solution it is necessary to constrain $\vec{\alpha}$. The common Prony algorithm imposes the constrain $\alpha_N = 1$.

To proceed, we addend the constraint (α_N-1) with a Lagrange multiplier (λ) to equation 2, and equate the derivatives with respect to α_i and λ to zero. Of course with the trivial constraint $\alpha_N = 1$, we could immediately reduce the dimensionality of $\vec{\alpha}$ by one, and equate the partial derivatives with respect to the remaining components of $\vec{\alpha}$ to zero. We also note that additional constraints, such as those constraining poles, can also be addended by additional Lagrange multipliers.

3. NEW PRONY ALGORITHM

It is important to note that the vector $\vec{\alpha}$ contains the coefficients of the characteristic polynomial. Since we are interested in the roots of this polynomial, the direction of the vector $\vec{\alpha}$ is the important criteria, and not its magnitude. In other words, once the proper direction of $\vec{\alpha}$ is determined, changing its magnitude will not alter the roots.

As was illustrated in the previous section, the common Prony algorithm searches for the direction of $\vec{\alpha}$ which minimizes the quadratic form of equation 2, while constraining the tip of the $\vec{\alpha}$ vector to be on the hyperplane $\alpha_N = 1$. Obviously then, in the search for the minimizing $\vec{\alpha}$, all directions cannot be covered.

A much better constraint on the $\vec{\alpha}$ vector is to constrain the tip of the $\vec{\alpha}$ vector to be on a hypersphere. In this case all directions are covered. Indeed, it is easy to see that such a constraint is the best of all possible since there can be no other direction which will yield a lower value for the quadratic form of equation 2.

The hypersphere constraint was mentioned by Golub in a private communication to Dudley [2], in the context that this constraint gave equal weight to all components of $\vec{\alpha}$, rather than stressing a single component. It was not pointed out, however, that this constraint will yield

the lowest possible minimum. Furthermore, the constraint was dismissed as being nonlinear and not particularly easy to apply.

A review of quadratic forms [4] indicates that the hypersphere constraint is about as easy to use in practice as the hyperplane constraint. For convenience, we choose a unit radius on the hypersphere, so that the constraint takes the form

$$\sum_{i=1}^{N} \alpha_i^2 - 1 = 0 . \tag{3}$$

We wish to minimize equation 2 subject to this constraint.

Using the Lagrange multiplier λ, we minimize

$$\vec{\alpha}^T A \vec{\alpha} - \lambda (\alpha_1^2 + \alpha_2^2 + \ldots + \alpha_N^2 - 1)$$

Accordingly we equate the partial derivatives with respect to the components of $\vec{\alpha}$ to zero; that is,

$$\frac{\partial}{\partial \alpha_i} \vec{\alpha}^T A \vec{\alpha} - 2\lambda\alpha_i = 0; \qquad i = 1,\ldots,N \tag{4}$$

But it is easily shown that

$$\frac{\partial}{\partial \alpha_i} \vec{\alpha}^T A \vec{\alpha} = 2(a_{i1}\alpha_1 + a_{i2}\alpha_2 + \ldots, a_{iN}\alpha_N),$$

so that equation 4 is

$$A \vec{\alpha} - \lambda \vec{\alpha} = \vec{0}$$

In other words, the λ values are the eigenvalues of matrix A.

At this point, we note that equation 2 is a non-negative, definite quadratic form, due to the nature of $(\vec{\delta}, \vec{\delta})$. Thus the eigenvalues of matrix A are all positive (≥ 0). Furthermore, the associated quadric surface is ellipsoidal.

Now multiply each member of equation 4 by α_i and add all the equations; the result is

$$\sum_{i=1}^{N} \alpha_i \frac{\partial}{\partial \alpha_i} \vec{\alpha}^T A \vec{\alpha} - 2\lambda(\alpha_1^2+\alpha_2^2+ \ldots+\alpha_N^2) = 0 . \qquad (5)$$

It is easy to show that the summation term is

$$\sum_{i=1}^{N} \alpha_i \frac{\partial}{\partial \alpha_i} \vec{\alpha}^T A \vec{\alpha} = 2 \vec{\alpha}^T A \vec{\alpha} .$$

Therefore, in light of the unit hypersphere constraint, equation 5 reduces to

$$\vec{\alpha}^T A \vec{\alpha} - \lambda = 0 . \qquad (6)$$

We have thus arrived at the following conclusions:

1. The extremes of $\vec{\alpha}^T A \vec{\alpha}$, subject to the constraint $|\vec{\alpha}| = 1$, are equal to the eigenvalues of matrix A. The lowest extreme (minimum value) is therefore equal to the lowest eigenvalue (λ_{min}).

2. The vector $\vec{\alpha}$ which causes this minimum value is the eigenvector corresponding to λ_{min}.

3. Simultaneously in finding λ_{min}, we have found the minimum value of the sum of the residuals squared.

Therefore, to solve the original minimization problem, we find the eigenvalues of $A = D^T D$ and the eigenvector corresponding to the minimum eigenvalue.

We note that the appearance of zero eigenvalues is correlated 1 to 1 with the rank of matrix D. That is, if the rank of D is N-k, there will be k zero eigenvalues (with unique eigenvectors). The appearance of small eigenvalues is thus associated with the order of the system, i.e., the number of exponentials.

4. CONSTRAINED POLES WITH THE NEW ALGORITHM

In the previous section, the coefficient vector $\vec{\alpha}$ which minimized the quadratic form $\vec{\alpha}^T A \vec{\alpha}$ with the constraint $|\vec{\alpha}| = 1$ was determined. The purpose here is to illustrate a method to determine the vector $\vec{\alpha}$ which minimizes the quadratic form under additional linear constraints. In particular, these linear constraints result from substituting specific poles (x) in the equation

$$\alpha_1 x^{N-1} + \alpha_2 x^{N-2} + \ldots + \alpha_N = 0 \; . \qquad (6)$$

Complex poles result in two, real, constraint equations; however, for two conjugate complex roots, only two equations are necessary.

In general, if there are K<N constraint equations, the linear constraints have the form

$$(\vec{c}_k, \vec{\alpha}) = 0 \quad ; \qquad k = 1, 2, \ldots, K.$$

The vector $\vec{\alpha}$, then, must lie in the intersection of the K planes which are orthogonal to the vectors $\vec{c}_k$.

The idea is to find a new set of orthonormal coordinate vectors such that K of them span the space occupied by the vectors $\vec{c}_k$. We then perform a coordinate transformation on the vector $\vec{\alpha}$ with the matrix formed by this new set of orthonormal vectors. If this matrix is designated by B, then the transformed vector $\vec{\alpha}$ (designated $\vec{\beta}$) is

$$\vec{\beta} = B\vec{\alpha} \Rightarrow \vec{\alpha} = B^{-1}\vec{\beta}$$

Thus, in the new coordinate system, the quadratic form is

$$\vec{\alpha}^T A \vec{\alpha} = (B^{-1}\vec{\beta})^T A B^{-1}\vec{\beta} \tag{7}$$

$$= \vec{\beta}^T (B^{-1})^T A B^{-1} \vec{\beta}$$

Since B is formed from orthonormal vectors,

$$B^{-1} = B^T ,$$

so that equation 7 can be written

$$\vec{\alpha}^T A \vec{\alpha} = \vec{\beta}^T BAB^T\vec{\beta} \quad . \tag{8}$$

In the new coordinate system, the constraints take the form of zero coordinates; that is, the vector $\vec{\beta}$ has K zero components. Therefore, a new quadratic form can be determined by setting the 1^{st} K rows and columns of BAB^T to $\vec{0}$. The eigenvalues of this new submatrix are determined, and the minimum found together with the corresponding eigenvector. This eigenvector is then re-transformed back to the original coordinate system, and the roots of the associated equation found.

In order to implement the above procedure, we must find the matrix B. Designate the vectors of B by $\vec{b}$. One way to find the $\vec{b}$'s is as follows:

$$\text{Let } \vec{b}_1 = \vec{c}_1 \ .$$

Then, another vector can be formed as a linear combination of $\vec{c}_1$ and $\vec{c}_2$; i.e.,

$$\vec{b}_2 = d\vec{c}_1 + \vec{c}_2 \ .$$

Furthermore, $\vec{b}_1$ and $\vec{b}_2$ must be orthogonal; therefore

$$(\vec{b}_1,\vec{b}_2) = 0 = d(\vec{c}_1,\vec{c}_1) + (\vec{c}_1,\vec{c}_2)$$

$$\Rightarrow d = -\frac{(\vec{c}_1,\vec{c}_2)}{(\vec{c}_1,\vec{c}_1)}$$

Thus, $\vec{b}_2$ is known. Similarly a 3^{rd} and higher order vectors can be found; i.e.,

$$\vec{b}_3 = e\vec{c}_1 + f\vec{c}_2 + \vec{c}_3 ,$$

$$(\vec{b}_1, \vec{b}_3) = 0$$

$$(\vec{b}_2, \vec{b}_3) = 0 \quad ,$$

and e and f can be found.

In this way, K orthogonal vectors can be found which span the space of the $\vec{c}_k$'s (constraint space).

The set of $\vec{b}$'s must be completed by the addition of N-K orthogonal vectors so that the complete set spans the space N. Again, one method to do this is as follows:

The additional vectors are determined by the relations

$$\left.\begin{aligned} (\vec{b}_1, \vec{b}_i) &= 0 \\ (\vec{b}_2, \vec{b}_i) &= 0 \\ &\vdots \\ (\vec{b}_{i-1}, \vec{b}_i) &= 0 \end{aligned}\right\} \quad i = K+1, K+2, \ldots, N$$

These relations must be successively solved for the components of $\vec{b}_i$ starting with i = K+1. In general, the set of equations are underdetermined, so that N-i-1 of the components are arbitrary. Of these arbitrary components, N-i-2 may be set equal to zero, which will aid in the solution process.

When all of the vectors $\vec{b}$ have been determined, they are normalized by dividing each by its magnitude. The matrix B is then formed using these vectors as rows, i.e.,

$$B = \begin{bmatrix} \vec{b}_1 \\ \vec{b}_2 \\ \vdots \\ \vec{b}_k \\ \vdots \\ \vec{b}_N \end{bmatrix} \quad \text{constraint space}$$

5. SOME NUMERICAL RESULTS WITH THE NEW ALGORITHM

In order to compare the new algorithm with the old, a computer program was written for both on a DTC microcomputer. To compare the two methods, a data set of 20 points was generated by contaminating the function

$$f(t) = \varepsilon^{-t} + \varepsilon^{-2t}$$

with random noise. The resulting data points are plotted in Figure 1. As can be seen from this data, the noise level is very high.

Table 1 is a direct comparison of the sum of the squares of the residuals between the old and the new methods for various extracted poles. As expected, in every case the new method "fits" the data better than the old.

Table 2 is another interesting comparison. Here, the sums of the squared differences between the original data and the "fitted" exponential model are shown. Householder [3] has proposed this criterion; however, in

order to determine the poles, an iterative sequence seems to be necessary. The new algorithm also tends to fit the data better with this criterion; however, it does not always do so. This is a clear demonstration that this criterion and the sum of the residuals squared are not the same.

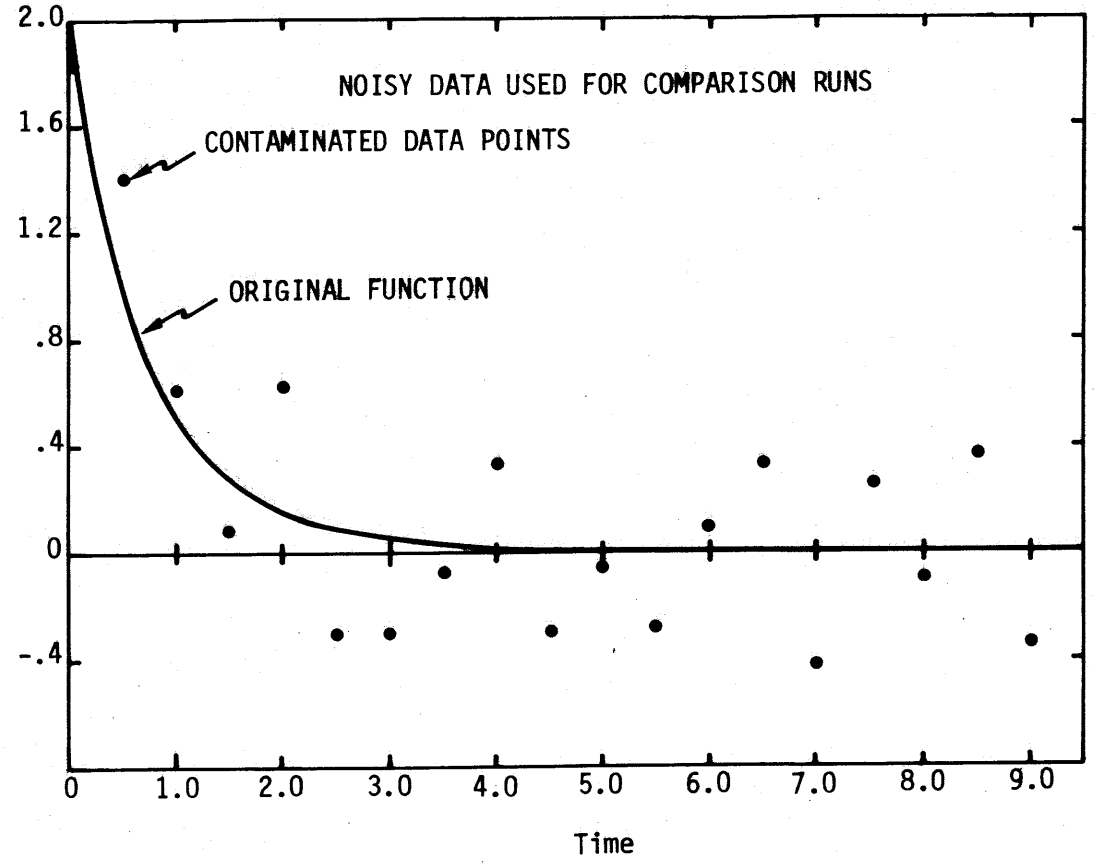
NOISY DATA USED FOR COMPARISON RUNS
CONTAMINATED DATA POINTS
ORIGINAL FUNCTION
2.0
1.6
1.2
.8
.4
0
-.4
0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
Time

Figure 1

TABLE 1

SUMS OF SQUARES OF RESIDUALS

Number of Poles	SUMS	
	Old Algorithm	New Algorithm
2	1.49871	1.393
3	1.16492	.951039
4	.885874	.642379
5	.470922	.36817
6	.330947	.237466

TABLE 2

SUMS OF SQUARES OF DIFFERENCES

Number of Poles	Old Algorithm	New Algorithm
2	1.83369	1.63449
3	1.34773	1.57332
4	1.48551	1.59467
5	1.98595	1.54028
6	.673739	.40347

6. CONCLUSIONS AND RECOMMENDATIONS

A reasonable simple scheme for implementing a new Prony algorithm has been devised. The new algorithm yields a sum of squared residuals equal to or lower than the usual algorithm.

A computer program using this algorithm should be written for a large digital computer, incorporating the pole constraints. The effects of noise, and the use of various smoothing and filtering schemes can then be determined.

REFERENCES

1. Van Blaricum, M. L., and R. Mittra, "Techniques for Extracting the Complex Resonances of a System Directly from its Transient Response," AFWL Interaction Note 301, December 1975.

2. Dudley, D. G., "Fitting Noisy Data with a Complex Exponential Series," AFWL Mathematics Note 51, March 1977.

3. Householder, A. S., On Prony's Method of Fitting Exponential Decay Curves and Multiple-Hit Survival Curves, Oak Ridge National Laboratory, ORNL-455, Oak Ridge, Tennessee, September 1949.

4. Guillemin, E. A., The Mathematics of Circuit Analysis, John Wiley and Sons, Inc., New York, 1949.