Fast Newton-Raphson Method for Moving-Average Spectral Factorization

Cédric J. DEMEURE and Clifford T. MULLIS

University of Colorado, Boulder, CO 80309, USA.

RESUME

Dans cet article, nous présentons une implémentation de la méthode de Newton-Raphson pour le calcul du facteur spectral à phase minimale d'une séquence de corrélation définie positive de dimension finie. L'avantage principal d'une telle méthode réside dans sa convergence quadratique. Chaque approximation successive consiste en la résolution d'un système d'équations linéaires. Deux algorithmes peuvent être utilisés pour résoudre ce système : le test de stabilité de Jury, ou une version symétrique de l'algorithme d'Euclid. Plusieurs propriétés de la méthode de Newton-Raphson sont étudiées. La méthode est finalement comparée dans son ensemble à d'autres méthodes classiques telles que la méthode de Durbin et celle de Bauer.

SUMMARY

In this paper, we present an implementation of the Newton-Raphson method to compute the minimum phase moving-average spectral factor of a finite positive definite correlation sequence. The major advantage of such a method is contained in its quadratic convergence behavior. Each step in the successive approximation method involves a system of linear equations that is solved using either the Levinson algorithm backwards (the Jury stability test), or a symmetrized version of the Euclid algorithm. Various properties of the Newton-Raphson map are studied. The overall efficiency of the method is then compared with the classical methods of Durbin and Bauer.

I. Introduction†

Given a finite symmetric positive definite discrete correlation sequence $\{r_i\}_{i=-n}^{n}$, the moving average spectral factorization problem consists of the computation of the moving-average (FIR) digital filter $H(z)$ of degree n, satisfying $R(z) = H(z)H(z^{-1})$. The computation consists of $(n+1)$ quadratic equations in the coefficients of the polynomial $H(z)$. Such a problem appears in computing Wiener filters, in system identification and data modelling. The solution is unique when minimum phase is required, which means that $H(z)$ has all its roots inside the unit circle. If the correlation sequence is not positive definite, then the problem is called a separation problem and alternative techniques and parametrizations must be used [11].

Several methods have been derived to solve this problem, The Bauer method [5] is based on the Cholesky factorization of a banded Toeplitz correlation matrix that is symmetric and semi-infinite. The non-zero elements in the rows of the Cholesky factor converge to the minimum phase solution (see proof and details in [7]). A faster algorithm to perform the factorization was introduced by Rissanen [6] and Bareiss [8]. This algorithm is linear in its complexity but also linear in its convergence behavior. Another method was proposed by Durbin. It is based upon the use of two autoregressive models in a double inversion technique [4], which is not iterative and so only gives an approximate MA factorization.

A Newton Raphson technique was proposed by Wilson [1], and extended to the matrix case by Arp [9]. It involves the solution of a linear set of equations for each update. The matrix involved in the system is called the Jury matrix. We have proposed a fast version of the Wilson algorithm, based on a factorization of the Jury matrix [2]. An even faster algorithm for solving the same system is presented in this paper. It consists in a polynomial representation and the use of the Euclid algorithm [10]. A "super-fast" $O(n(log_2 n)^2)$ implementation of the Euclid algorithm is even possible [12].

† This work was supported by the Army Research Office, Research Triangle Park, NC under contract DAAG 29-84-K-0014, and by the Office of Naval Research, Arlington, VA under contract N00014-85-K-0256.

This paper is organized as follows. First, we review the Newton-Raphson method for solving this non-linear problem, as well as some of its properties. We derive then a fast algorithm to compute the Newton-Raphson step based on a factorization of the Jury matrix. That factorization is obtained by applying the Levinson algorithm backwards on the filter coefficients. The complexity of that approach is $O(2n^2)$ operations per iteration, which is an order of magnitude lower than the $O(n^3/3)$ operations needed in a standard Gauss elimination procedure. We introduce an alternative polynomial approach and show how a symmetrized version of the Euclid algorithm may also be used to solve the problem. The complexity falls down to $O(3n^2/2)$ operations, a 25% gain compared to the matrix presentation. The memory requirements of the algorithm become linear in n whereas they were quadratic in both the matrix presentation and the Gauss elimination procedure. Finally, we show some experimental results and compare the overall complexity of the Newton method with traditional methods, such as the Durbin and Bauer methods.

Notations

$\mathbf{h}$ is a row vector of length n, h_i is its i^{th} component, and $\mathbf{h}(t)$ is its t^{th} approximation. $H(z)$ is a polynomial in z, a complex variable, and $H_t(z)$ is the t^{th} approximation of $H(z)$.

II. The Newton-Raphson Method

Given the set of n correlation values $\{r_i\}_{i=-n}^{n}$, the finite symmetric correlation sequence has z-transform

$$R(z) = R(z^{-1}) = r_0 + \sum_{i=1}^{n} r_i(z^n + z^{-n})$$

Let's represent $R(z)$ by

$$R(z) = \mathbf{r}\theta(z)$$

where $\mathbf{r} = [r_0/2, r_1, \ldots, r_n]$, $\theta(z) = \psi(z) + \psi(z^{-1})$, and $\psi(z) = [1, z^{-1}, z^{-2}, \ldots, z^{-n}]^T$. In terms of the filter coefficients, $\mathbf{r} = f(\mathbf{h})$, where $f(\mathbf{h})$ is a quadratic function in the coefficients of $\mathbf{h}$. The quadratic function may be written $f(\mathbf{h}) = \mathbf{h}F(\mathbf{h})$, where $F(\mathbf{h})$ is an $(n+1)$ by $(n+1)$ matrix.

In a successive approximation scheme, one begins with the

approximation $H_t(z) = \sum_{i=0}^{n} h_i(t) z^{-i}$, with

$$\mathbf{h}(t) = [h_0(t), h_1(t), \ldots, h_n(t)],$$

and builds the error vector

$$\mathbf{e}(t) = f(\mathbf{h}(t)) - \mathbf{r} = \mathbf{h}(t)F(\mathbf{h}(t)) - \mathbf{r}.$$

The Newton-Raphson map is then defined by $\mathbf{h}(t+1) = \mathbf{h}(t) - \mathbf{e}(t)S^{-1}(\mathbf{h}(t))$, where the matrix $S(\mathbf{h}(t))$ contains the partial derivatives of $\mathbf{e}(t)$ with respect to $\mathbf{h}(t)$:

$$S(\mathbf{h}(t))_{i,j} = \frac{\partial e_i(t)}{\partial h_j(t)} \qquad \forall i,j \in [0,n].$$

It is easily shown that $S(\mathbf{h}(t)) = 2F(\mathbf{h}(t))$, so that the Newton-Raphson iteration is given by

$$\begin{aligned}\mathbf{h}(t+1) &= \mathbf{h}(t) - (\mathbf{h}(t)F(\mathbf{h}(t)) - \mathbf{r})\left[2F(\mathbf{h}(t))\right]^{-1} \\ &= \frac{1}{2}\left(\mathbf{h}(t) + \mathbf{r}F^{-1}(\mathbf{h}(t))\right)\end{aligned}$$

which is the vector equivalent of the scalar "method of averaging": $x(t+1) = (x(t) + d/x(t))/2$ to compute the square root $x = \sqrt{d}$.

Some properties of the Newton-Raphson relation are:

1) $\mathbf{a}F(\mathbf{b}) = \mathbf{b}F(\mathbf{a})$, for all row vectors $\mathbf{a}, \mathbf{b}$.

2) Post-multiply the matrix $S(\mathbf{h}(t))$ by $[\psi(z) + \psi(z^{-1})]$:

$$S(\mathbf{h}(t))\left[\psi(z) + \psi(z^{-1})\right] = H_t(z)\psi(z^{-1}) + H_t(z^{-1})\psi(z)$$

and use the map to get

$$\begin{aligned}H_t(z)H_{t+1}(z^{-1}) + H_t(z^{-1})H_{t+1}(z) = \\ H_t(z)H_t(z^{-1}) + R(z)\end{aligned} \tag{1}$$

which is another way to characterize the map between $H_t(z)$ and $H_{t+1}(z)$.

3) By rearranging (1), we have

$$\begin{aligned}H_{t+1}(z)H_{t+1}(z^{-1}) = \\ [H_t(z) - H_{t+1}(z)][H_t(z^{-1}) - H_{t+1}(z^{-1})] + R(z)\end{aligned}$$

This equation together with the Rouché theorem are useful to show the convergence of the method. The minimum phase property of $H_{t+1}(z)$, given the minimum phase property of $H_t(z)$ may also be proven. It also shows that the method is self-correcting. These conclusions are only valid if the actual computation error is not big enough, so that zeros do not cross the unit circle (See [1] for proof and details).

4) If $\delta(t)$ is the maximum absolute value of the elements of the error row vector (difference between $\mathbf{h}(t)$ and the exact solution $\mathbf{h}$), at step t then

$$\delta(t+1) = M(n+1)(\delta(t))^2$$

where M is a constant, and $\delta(t)$ small enough. This means that convergence is quadratic near the solution [1].

5) The extreme points on the spectrum behave as

$$H_{t+1}(1) = \frac{1}{2}\left[H_t(1) + \frac{R(1)}{H_t(1)}\right]$$

$$H_{t+1}(-1) = \frac{1}{2}\left[H_t(-1) + \frac{R(-1)}{H_t(-1)}\right]$$

which means that they behave as in the scalar case, so that if the initial guess is such that $H_0(1) = \sqrt{R(1)}$ and $H_0(-1) = \sqrt{R(-1)}$ then those points will match from the beginning and will not change.

6) If $H_t(z)$ is minimum phase then $h_0(t+1) \geq h_0(t)/2$, with equality if and only if $R(z) \equiv 0$. If in addition $|H_t(z)|^2 \geq R(z)$ on the unit circle, then $h_0(t+1) \leq h_0(t)$. To prove these properties, take equation (7), divide it by $H_t(z)H_t(z^{-1})$ and integrate on the unit circle :

$$2\frac{h_0(t+1)}{h_0(t)} = 1 + \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{R(e^{j\theta})}{|H_t(e^{j\theta})|^2} d\theta.$$

8) If $H_t(z)$ is minimum phase, $|H_t(z)|^2 \geq R(z)$ on the unit circle, and $h_0(t+1) = h_0(t)$, then $H_{t+1}(z) = H_t(z)$ and $H_t(z)H_t(z^{-1}) = R(z)$, which means that we can test convergence by looking at how the first component of the row vector behaves.

III. The Jury Matrix

The Jury matrix $S(\mathbf{h})$ is defined by [3]:

$$S(\mathbf{h})_{i,j} = \begin{cases} h_i & \text{if } j = 0, \\ h_{i+j} + h_{i-j} & \text{otherwise.}\end{cases}$$

where h_i is set to zero outside the range $[0,...,n]$. This matrix also arises in a different context, in writing the normal (Yule-Walker) equations for an autoregressive sequence, whose correlation sequence is r_t:

$$\mathbf{r}S(\mathbf{a})^T = \mathbf{a}R = \sigma^2[1,0,\ldots,0]^T$$

where R is the symmetric Toeplitz correlation matrix which top row is $[r_0,\ldots,r_n]$. This means that this matrix is useful when computing the auto-correlation sequence r_i from the auto-regressive (AR) filter parameters and also when computing the auto-correlation sequence form the autoregressive moving-average (ARMA) filter coefficients [10].

Eigenvalues The set of eigenvalues may be studied in a rather simple way : if λ is such that $H(\lambda) = H(\lambda^{-1})$, then $H(\lambda)$ is an eigenvalue, since

$$S(\mathbf{h})[\psi(\lambda) + \psi(\lambda^{-1})] = H(\lambda)\psi(\lambda^{-1}) + H(\lambda^{-1})\psi(\lambda).$$

In particular, $H(1)$ and $H(-1)$ are eigenvalues of $S(\mathbf{h})$. If $h_i = 0$, $m < i \leq n$, then $h_0 = H(\infty)$ is an eigenvalue of order at least $n-m$, as the matrix $S(\mathbf{h})$ becomes partially triangular. In summary, the matrix has one eigenvalue at $H(1)$, another at $H(-1)$, another of multiplicity $n-m$ at h_0 and the others are of the type $H(\lambda) = H(1/\lambda)$.

Determinant The determinant of $S(\mathbf{h})$ may be shown to be equal to [2]

$$\det[S(\mathbf{h})] = h_0^{n+1}\prod_{i=1}^{n}\prod_{j=1}^{i}(1 - \alpha_i\alpha_j),$$

where the coefficients α_i are the roots of $H(z)$. This result allows us to predict that if a root of $H(z)$ is close to the unit circle, then the problem is ill-conditioned and convergence will be slow.

Factorization The filter coefficients may be coded with the reflection coefficients $k_1, k_2, \ldots, k_n$. These reflection coefficients are determined by running the Levinson recursions backwards:

$$\mathbf{h}^n = [\mathbf{h}^{n-1}, 0] + k_n[0, \mathbf{h}^{n-1}J] \tag{2}$$

where J is the exchange matrix (containing ones on its main anti-diagonal). Here superscripts denote the degree of the associated polynomial. Then the Jury matrix $S(\mathbf{h}^n)$ satisfies the recursion:

$$S(\mathbf{h}^n) = Q_n \left[\begin{array}{ccc|c} & S(\mathbf{h}^{n-1}) & & \begin{matrix}0\\ \vdots \\ 0\end{matrix} \\ \hline 0 & h_{n-1}^{n-1} \quad \ldots \quad h_1^{n-1} & & h_0^{n-1}\end{array}\right]$$

with

$$Q_n = \begin{bmatrix} 1 & & & k_n \\ & \ddots & \iddots & \\ & \iddots & \ddots & \\ k_n & & & 1 \end{bmatrix}$$

This procedure may be repeated to produce the factorization $S(\mathbf{h}) = Q_nQ_{n-1}\ldots Q_1L$, where

$$L = \begin{bmatrix} h_0^0 & & & & \\ 0 & h_0^0 & & & \\ 0 & h_1^1 & h_0^1 & & \\ \vdots & \vdots & & \ddots & \\ 0 & h_{n-1}^{n-1} & \ldots & h_1^{n-1} & h_0^{n-1} \end{bmatrix}$$

In Q_i, the matrix J has dimension $(i+1)$ by $(i+1)$. The Q_i matrix has inverse

$$Q_i^{-1} = \begin{bmatrix} \frac{1}{(1-k_i^2)}(I - k_iJ) & 0 \\ 0 & I \end{bmatrix}$$

To compute $\mathbf{x}(t) = \mathbf{h}(t+1) - \mathbf{h}(t)/2 = \mathbf{r}S^{-1}(\mathbf{h}(t))$, one just needs to solve a triangular system of n equations. The factorization induces another formula for the determinant of the Jury matrix in terms of the variables of the algorithm (the reflection coefficients):

$$\text{Det } Q_i = \begin{cases} (1-k_i^2)^j, & \text{if } i = 2j, \\ (1-k_i^2)^j(1+k_i), & \text{if } i = 2j+1. \end{cases}$$

so that

$$\text{Det } S(\mathbf{h}) = (h_0)^{n+1} \prod_{i=1}^{n} \text{Det } Q_i$$

as $h_0^i = h_0^n = h_0$. This result was first found by Barndorff-Nielsen and Shou [13]. The closer k_i is to 1 or -1, the more unstable the algorithm, or equivalently the more ill-conditioned is the system of equations. Instability may then be detected in real time, before it occurs. The algorithm to solve the Jury system of equations has a complexity of $2n^2$ operations and a memory requirement of $n^2/2$. The quadratic memory requirements may be dropped at the expense of building the lower degree polynomials when needed, at a complexity of $n^2/2$ operations.

IV. Symmetrized Euclid Algorithm

In this section we introduce an alternative algorithm to compute the Newton-Raphson step, based on a polynomial treatment. The system of equations (1) used in the previous presentation is equivalent to the polynomial equation

$$H_t(z)X(z^{-1}) + H_t(z^{-1})X(z) = E(z) + E(z^{-1}) \qquad (3)$$

with $X(z) = H_{t+1}(z) - H_t(z)/2$ and

$$E(z) = r_0/2 + \sum_{i=1}^{n} r_i z^{-i}.$$

The algorithm to solve (3) consists of two steps per degree : one step to reduce the degree of the right hand side, and one step to reduce the degree of $H_t(z)$. The first step consists of replacing $E(z) = E_n(z)$ by $E_{n-1}(z)$ such that

$$E_{n-1}(z) = E_n(z) - \alpha_n z^{-n} H_t(z^{-1})$$

where the variable α_n is chosen so that the degree of $E_{n-1}(z)$ is strictly less than the degree of $E_n(z)$. Equation (3) is modified to

$$\begin{aligned} H_t(z)\left[X(z^{-1}) - \alpha_n z^n\right] + H_t(z^{-1})\left[X(z) - \alpha_n z^{-n}\right] \\ = E_{n-1}(z) + E_{n-1}(z^{-1}) \end{aligned} \qquad (4)$$

The second step consists of replacing the degree n polynomial $H_t(z) = G_n(z)$ by the degree $n-1$ polynomial $G_{n-1}(z)$ (the Jury algorithm, or Levinson's backwards):

$$G_{n-1}(z) = G_n(z) - k_n z^{-n} G_n(z^{-1}) \qquad (5)$$

Equation (5) is just another way to write equation (2). The transformation between $G_n(z)$ and $G_{n-1}(z)$ may be written

$$\begin{bmatrix} G_n(z) \\ G_n(z^{-1}) \end{bmatrix} = \frac{1}{1-k_n^2} \begin{bmatrix} 1 & k_n z^{-n} \\ k_n z^n & 1 \end{bmatrix} \begin{bmatrix} G_{n-1}(z) \\ G_{n-1}(z^{-1}) \end{bmatrix}$$

Use this reduction in equation (32) to get

$$\begin{aligned} G_{n-1}(z)X_{n-1}(z^{-1}) + G_{n-1}(z^{-1})X_{n-1}(z) \\ = E_{n-1}(z) + E_{n-1}(z^{-1}) \end{aligned}$$

where $X(z) = X_n(z)$ and

$$\begin{bmatrix} X_n(z) \\ X_n(z^{-1}) \end{bmatrix} = \alpha_n \begin{bmatrix} z^{-n} \\ z^n \end{bmatrix} + \begin{bmatrix} 1 & -k_n z^{-n} \\ -k_n z^n & 1 \end{bmatrix} \begin{bmatrix} X_{n-1}(z) \\ X_{n-1}(z^{-1}) \end{bmatrix}$$

The same process is then iterated with decreasing degree. In summary, we have the following symmetrized Euclid algorithm :

Down Steps For $i = n, \ldots, 1$:

$$G_{i-1}(z) = G_i(z) - k_i z^{-i} G_i(z^{-1})$$
$$E_{i-1}(z) = E_i(z) - \alpha_i z^{-i} G_i(z^{-1})$$

Solution $X_0(z) = E_0(z)$

Up Steps For $i = 1, \ldots, n$:

$$X_i(z) = X_{i-1}(z) + z^{-i}\left[\alpha_i - k_i X_{i-1}(z^{-1})\right]$$

This algorithm requires $O(3n^2/2)$ operations and linear memory storage $O(5n)$, which is an improvement compared to the previous matrix method. Note that the main saving is obtained by the use of both the factorization of the Jury matrix and the simultaneous reduction of the right hand side.

V. Experimetal Results

We have implemented the Newton-Raphson fast algorithm on three examples in order to compare the results with several other techniques. The error measure at step t is defined to be

$$e(t) = \sum_{i=0}^{n} (h_i(t) - h_i)^2$$

where the exact coefficients are $\{h_i\}_{i=0}^n$, and $\{h_i(t)\}_{i=0}^n$ are the coefficients of the t^{th} approximation. The exact filter in our examples is:

$$H(z) = \prod_{i=1}^{n} (1 - z_i z^{-1})$$

where

1) $n = 4, \quad z_i = e^{j\theta_i}, \quad \theta_i \in (\pm 60^o, \pm 140^o)$

2) $n = 8, \quad z_i = .9e^{j\theta_i}, \quad \theta_i \in (\pm 25^o, \pm 90^o, \pm 155^o)$

3) $n = 8, \quad H(z) = 1 + .95z^{-8}$

Convergence is then reached when the error $e(t)$ is less than a threshold ε. To compare the performance of the Newton-Raphson method, we ran this example with several methods:

1) Durbin [4] This algorithm is based on two autoregressive models, the first one to fit the original correlation sequence, zero-padded up to L ($L \gg n$), and the second to fit the correlation of the parameters of the first. The two autoregressive factorization problems are computed using the Levinson algorithm. This method is not a successive approximation method, as the only freedom is in the choice of L. The minimum phase property of $C(z)$ is ensured.

2) Bauer [5] Compute successive rows of H, the Cholesky factor for the symmetric banded Toeplitz matrix $R = HH^T$. H is banded and lower triangular. If the matrix R is extended to become semi-infinite, the non-zero terms in the rows of H converge to the constant solution to the problem. Convergence and minimum phase are ensured [7].

3) Rissanen [6] This is a fast version of the previous algorithm, which uses the Toeplitz property of R in a more efficient way. It computes the same parameters as the Bauer algorithm at each iteration, but with fewer internal variables and operations.

Figures 1, 2, and 3 show the magnitude of the frequency response for estimates of $H(z)$ at various steps of each algorithm, for the first example. The magnitude responses are plotted on a logarithmic scale. Figure 1 shows these plots for the Durbin method when the values for the long AR model order L are equal to 1,2,4,8,... Figure 2 shows the same plots for the method of Bauer and Rissanen, and the steps plotted also correspond to a geometric progression: 1,2,4,8,... Figure 3 shows the same plots for the Newton method and every single step is plotted. The logarithmic scale has been chosen to emphasize the behavior of the algorithms at the polynomial zeros (which are on the unit circle for that example).

Table 1 compares these algorithms in terms of their computational complexity per iteration, memory requirements, and convergence behavior. Note that the Bauer algorithm is more expensive to use than the Rissanen algorithm, as is expected. If the necessary value of L in the Durbin method is too large, then it is also very expensive. With $\varepsilon = 10^{-8}$ for the first example, Table 2 gives the number of floating point operations necessary to reach an error less than or equal to ε. The x in Table 2 is due to the fact that the Levinson algorithm was unable to lower the error below the threshold in a reasonable amount of computation ($L < 2000$). The fast Cholesky algorithm (Rissanen) can be made faster than the Newton-Raphson algorithm with the use of a parallel vector machine. The number of iterations (8 for Newton and 361 for Rissanen in the first example) suggests that the Newton method will have better behavior when implemented using finite arithmetic, as the error will not accumulate as much. The Newton method uses the data r_i at each iteration, and the Rissanen algorithm uses it only at the beginning, meaning the later method is not a self correcting method.

VI. Conclusions

From these theoretical and computational results, the Newton method, implemented with a fast algorithm, appears to be a very efficient and stable way to perform the moving average factorization of a finite correlation sequence. Note also

that this factorization is equivalent to factoring a symmetric polynomial into its square roots.

VII. References

[1] G. T. Wilson, "Factorization of the Covariance Generating Function of a Pure Moving Average Process," SIAM J. Numerical Analysis, Vol. 6, No. 1, pp 1-7, March 1969.

[2] C.T. Mullis and C.J. Demeure, "The Jury Matrix and a Newton-Raphson Procedure for MA Spectral Factorization," Proceedings of the XX^{th} Asilomar Conf. on Sig., Syst. and Comp., Pacific Grove, Nov. 1986.

[3] E. Jury, *Theory and Application of the z-transform Method*, Wiley Eds., New-York, 1964, page 297.

[4] J. Durbin, "Efficient Estimation of Parameters in Moving-Average Models," Biometrika, Vol. 46, pp 306-316, Dec. 1959.

[5] F. Bauer, "Ein Direktes Iterationsverfahren zur Hurwitz-Zerlegung eines Polynoms," Archiv. der Electrischen Ubertragung, Vol. 9, No. 6, pp 285-290, June 1955.

[6] J. Rissanen, "Algorithms for Triangular Decomposition of Block Hankel and Toeplitz Matrices with Application to Factoring Positive Matrix Polynomials," Math. of Computation, Vol. 27, No. 121, pp 147-154, Jan. 1973.

[7] J. LeRoux, "Sur les Algorithmes de Factorization Spectrale dans le cas des Signaux Stationnaires Echantillonnés," Thèse de Doctorat d'Etat, Université de Nice, France, Oct. 1985.

[8] E. Bareiss, "Numerical Solution of Linear Equations with Toeplitz and Vector Toeplitz Matrices," Numer. Math., Vol. 13, pp 404-424, 1969.

[9] F. Arp, "Iterative Algorithm for Time Discrete Spectral Factorization," Archiv. der Electrischen Übertragung, Band 39, Heft 6, pp. 251-258, 1985.

[10] C.J. Demeure and C.T. Mullis, "Euclid's Algorithm and the Fast Computation of Cross-Covariance and Auto-Covariance Sequences," To appear in IEEE Trans. on ASSP, april 1989.

[11] G.O. Feyh and C.T. Mullis, "Moving Average Separation," IEEE Proc. Int. Conf. on ASSP, New York, pp. 2280-2283, April 1988.

[12] R.E. Blahut, *Fast Algorithms for Digital Signal Processing*, Addison Wesley, Reading, MA, 1985.

[13] O. Barndorff-Nielsen and G. Shou, "On the Parametrization of Autoregressive Models by Partial Autocorrelations," J. Multivariate Analysis, Vol. 3, pp. 408-419, 1973.

Algorithms	# Mult/Div	Memory	Behavior
DURBIN	$L(L+n)$	L	
BAUER	$n(n+5)/2$	$n(n+1)/2$	Linear
RISSANEN	$2n+2$	$2n$	Linear
NEWTON	$3n^2/2$	$5n$	Quadratic

Table 1: Comparison of Algorithms for Moving Average Spectral Factorization.

Algorithms	First Ex.	Second Ex.	Third Ex.
DURBIN	x	3240	622505
BAUER	6624	1353	18460
RISSANEN	3630	547	6390
NEWTON	192	216	672

Table 2: Number of Operations for the Moving Average Factorization Algorithms.

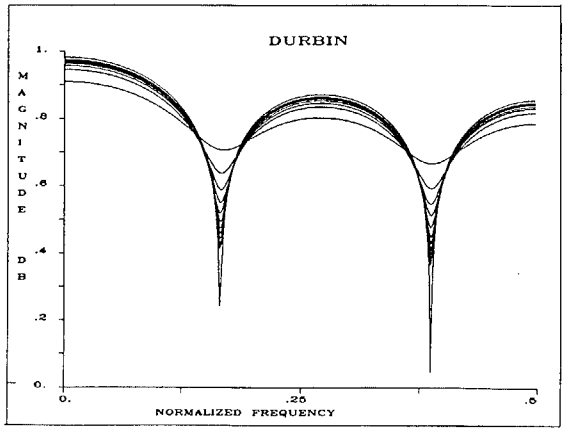

Figure 1: Frequency Domain Behavior of the Durbin Method.

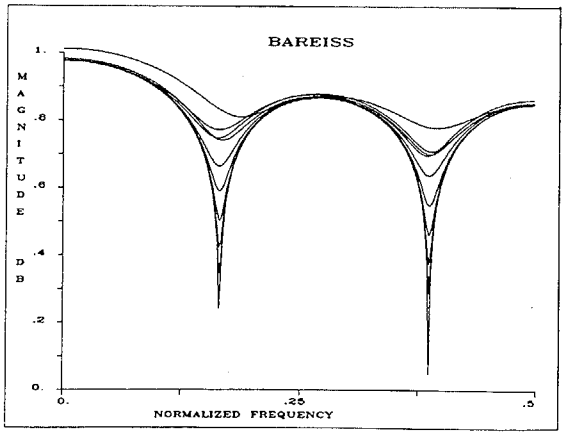

Figure 2: Frequency Domain Behavior of the Rissanen, Bareiss, Bauer Method.

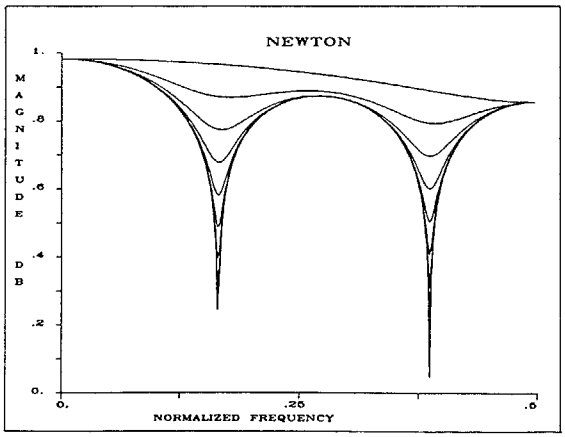

Figure 3: Frequency Domain Behavior of the Newton Method.