A Comparison of Numerator Estimators for ARMA Spectra

RANDOLPH L. MOSES AND A. A. (LOUIS) BEEX

Abstract—**This correspondence investigates the problems of estimating the numerator spectrum corresponding to an ARMA time series model once the denominator spectrum (i.e., the AR coefficients) has been estimated. A general form for an estimator of the numerator spectral (NS) coefficients is developed first. Six NS estimators from the recent literature are then compared by fitting them into this general framework and extracting their particular characteristics. It is shown that some methods are special cases of other methods, and that several of these methods are asymptotically equivalent.**

I. Introduction

An important engineering problem is to determine a model for a stochastic time series from a finite set of measurements $\{x(1), x(2), \cdots, x(n)\}$. One popular model is the autoregressive moving average (ARMA) model given by

$$x(k) + \sum_{i=1}^{p} a_i x(k-i) = \sum_{j=0}^{q} b_j \epsilon(k-j) \tag{1.1}$$

where $\{\epsilon(k)\}$ is zero mean, unit variance white noise. The associated power spectral density function for $\{x(k)\}$ is given by

$$S_x(z) = \frac{B(z)\,B^*(1/z^*)}{A(z)\,A^*(1/z^*)} = \frac{C(z)}{A(z)\,A(1/z^*)} \tag{1.2}$$

where

$$\begin{aligned} C(z) &= B(z)\,B^*(1/z^*) \\ &= c_{-q}z^{-q} + \cdots + c_0 + c_1 z + \cdots + c_q z^q. \end{aligned} \tag{1.3}$$

We shall refer to the c_k coefficients as the coefficients of the numerator spectrum (NS).

The ARMA modeling problem is to estimate the a_i (AR) and b_k (MA) coefficients from the measurements. One popular class of algorithms entails first estimating the AR coefficients, then using the AR estimates along with the given data measurements to estimate the NS coefficients. If necessary for a given application, the MA coefficients are recovered from the NS coefficients by performing a spectral factorization via (1.3).

This correspondence focuses on the estimation of the NS coefficients. Several methods for estimating the NS coefficients have been reported [1]–[7]. They are generally derived in different ways, and as a result it has been difficult to compare or contrast these methods. This correspondence makes such a comparison by first developing a general form for estimators of NS coefficients. We then compare six NS estimators by showing how they fit into this general form.

II. The General Numerator Spectrum Estimator

A. Exact Properties

To motivate the general form of the NS estimator, we first assume that the exact autocorrelation sequence and the exact AR coefficients for an ARMA (p, q) data sequence are given. Define the $(p+1) \times 1$ vector of AR coefficients as

$$\boldsymbol{a} = [a_0 a_1 a_2 \cdots a_p]', \quad a_0 \equiv 1. \tag{2.1}$$

Manuscript received May 6, 1985; revised May 13, 1986.

R. L. Moses is with the Department of Electrical Engineering, Ohio State University, Columbus, OH 43210.

A. A. Beex is with the Department of Electrical Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061.

IEEE Log Number 8610154.

From (1.1) we can define the forward prediction error process $\{f(k)\}$ by

$$f(k) = \sum_{i=0}^{p} a_i x(k-i) = \sum_{j=0}^{q} b_j \epsilon(k-j). \tag{2.2}$$

From (1.2) and (2.2) it can be seen that $C(z)$ is the power spectral density function corresponding to $\{f(k)\}$, so the c_k coefficients are just the autocovariances of $\{f(k)\}$

$$c_k = E\{f(m+k)\,f^*(m)\}. \tag{2.3}$$

Substituting (2.2) into (2.3) twice yields

$$c_k = \sum_{i=0}^{p} \sum_{j=0}^{p} a_i a_j^* \rho(k-i+j) \tag{2.4}$$

where $\{\rho(k)\}$ is the autocovariance sequence associated with $\{x(k)\}$, i.e.,

$$\rho(k) = E\{x(k+n)\,x^*(n)\}.$$

The Yule–Walker equations for an ARMA (p, q) process imply that $\sum_{i=0}^{p} a_i \rho(k-i+j) = 0$ for $j > q-k$, so c_k can alternatively be found by

$$c_k = \sum_{i=0}^{p} \sum_{j=0}^{q-k} a_i a_j^* \rho(k-i+j). \tag{2.5}$$

Equations (2.4) and (2.5) are not new and can be found in [1] and [2] among other places.

A more compact way to write (2.4) and (2.5) is

$$c_k = \boldsymbol{a}^H R_k \boldsymbol{a}, \quad -q \le k \le q, \tag{2.6}$$

where R_k is a $(p+1) \times (p+1)$ matrix given by

$$R_k = \begin{bmatrix} \rho(k) & \rho(k-1) & \cdots & \rho(k-p) \\ \rho(k+1) & \rho(k) & \cdots & \rho(k-p+1) \\ \vdots & \vdots & \ddots & \vdots \\ \rho(k+p) & \rho(k+p-1) & \cdots & \rho(k) \end{bmatrix} \tag{2.7}$$

for (2.4), and given by

$$R_k = \begin{bmatrix} \rho(k) & \rho(k-1) & \cdots & \rho(k-p) \\ \vdots & & & \vdots \\ \rho(q) & \rho(q-1) & \cdots & \rho(q-p) \\ 0 & 0 & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \tag{2.8}$$

for (2.5).

B. A General Estimator and Its Properties

Equation (2.6) motivates a form for a numerator estimator when only estimates of $\boldsymbol{a}$ and $\rho(k)$ are available. An estimate of c_k is obtained by replacing the exact quantities in (2.6) by the estimated ones. For generality, a windowing sequence $\hat{w}_k$ is also incorporated, giving

$$\hat{c}_k = \hat{w}_k \hat{\boldsymbol{a}}^H \hat{R}_k \hat{\boldsymbol{a}} \tag{2.9}$$

where we use the "$\hat{}$" to denote estimates based on the n data $x(1), \cdots, x(n)$. The windowing sequence $\{\hat{w}_k\}$ may be used to force $\{\hat{c}_k\}$ to be nonnegative definite (NND). Equation (2.9) represents the general form of an NS estimator.

From the properties of $\hat{\boldsymbol{a}}$, $\hat{w}_k$, and $\hat{R}_k$ in (2.9), we can answer several questions about the numerator estimator, such as:

1) what is the asymptotic bias and variance of the numerator estimate?

0096-3518/86/1200-1668$01.00 © 1986 IEEE

2) is $\{\hat{c}_k\}$ an NND sequence?

3) is $\hat{R}_k$ of the form (2.7) or (2.8)?

Each of these characteristics is discussed below.

1) Asymptotic Bias and Variance: It is not difficult to show that if $\hat{R}_k \overset{wp1}{\to} R_k$, $\hat{a}_k \overset{wp1}{\to} a_k$, and $\hat{w}_k \overset{wp1}{\to} w_k$, as $n \to \infty$, where "$\overset{wp1}{\to}$" denotes convergence is with probability 1, then

$$\hat{c}_k \overset{wp1}{\to} w_k c_k.$$

The convergence assumptions on $\hat{R}_k$, $\hat{a}_k$, and $\hat{w}_k$ are nearly always satisfied. Thus, the $\hat{c}_k$ estimates are asymptotically biased (by a known amount w_k), and the variances of the estimates approach zero. If $w_k = 1$, then the estimates are asymptotically unbiased and consistent.

2) Nonnegative Estimates: To obtain unbiased $\hat{c}_k$ estimates, it is preferable to choose $\hat{w}_k = 1$, or at least have $\hat{w}_k \to 1$. However, for this choice of $\hat{w}_k$, $\{\hat{c}_k\}$ is, in general, not an NND sequence (which implies that a spectral factorization of $C(z)$ will not yield a $B(z)$ that satisfies $B(e^{j\omega}) = B^*(e^{-j\omega})$). However, one can guarantee NND estimates for finite n (not just asymptotically) with a special choice of $\hat{w}_k$ and $\hat{R}_k$ in (2.9). Consider an $\hat{R}_k$ of the form

$$\hat{R}_k = \frac{1}{J} X_{s,t}^H X_{s+k,t+k} \tag{2.10}$$

where J is some normalizing constant independent of k, and where

$$X_{s,t} = \begin{bmatrix} x(s) & x(s-1) & \cdots & x(s-p) \\ x(s+1) & x(s) & \cdots & x(s+1-p) \\ \vdots & \vdots & & \vdots \\ x(t) & x(t-1) & \cdots & x(t-p) \end{bmatrix} \tag{2.11}$$

with the convention that $x(k) \equiv 0$ for $k < 1$ and $k > n$. In this case, (2.9) can be written as

$$\hat{f}_k = \sum_{i=0}^{p} \hat{a}_i x(k-i) \tag{2.12}$$

$$\hat{c}_k = \hat{w}_k \frac{1}{J} \sum_{m=s}^{t} \hat{f}_{m+k} \hat{f}^*_m. \tag{2.13}$$

Equations (2.13) and (2.12) can be interpreted as first passing the data $x(k)$ through the filter

$$\hat{A}(z) = 1 + \hat{a}_1 z^{-1} + \cdots + \hat{a}_p z^{-p}$$

to produce $\hat{f}_k$, then forming $\hat{c}_k$ as an autocovariance estimate of $\hat{f}_k$. This estimator produces NND $\hat{c}_k$ estimates for any NND window sequence, as shown in the following theorem.

Theorem: If $\hat{R}_k$ is given by (2.10) and if $\{\hat{w}_k\}_{k=-\infty}^{\infty}$ is supported on $|k| \le q$ and is NND, then $\{\hat{c}_k\}_{k=-\infty}^{\infty}$ is NND for any AR estimate $\hat{\boldsymbol{a}}$.

Proof: From (2.13) we can write

$$\hat{c}_k = \hat{w}_k \cdot \hat{v}_k$$

where the sequences are defined for $-\infty < k < \infty$. Since $\{\hat{v}_k\}$ is a constant times the convolution of $\{\hat{f}_m\}$ with $\{\hat{f}_{-m}\}$ (where $\hat{f}_k \equiv 0$ for $k < s$ or $k > t$), it is an NND sequence. It follows that $\hat{c}_k$ is NND. This is true because the Fourier transforms of $\{\hat{w}_k\}$ and $\{\hat{v}_k\}$ are both nonnegative and real, so their convolution is nonnegative and real. The latter is the Fourier transform of $\hat{w}_k \cdot \hat{v}_k$, so this sequence is NND. ■

The major purpose of the windowing sequence $\{\hat{w}_k\}$ is to ensure that $\{\hat{c}_k\}$ is NND. Two possible window sequences are the triangular window

$$\hat{w}_k = 1 - \frac{|k|}{L}, \quad L \ge q+1, \quad |k| \ge q \tag{2.14}$$

and the exponential window

$$\hat{w}_k = \alpha^{|k|}, \quad 0 < \alpha \le 1, \quad |k| \le q \tag{2.15}$$

with $\hat{w}_k \equiv 0$ for $|k| > q$. Since (2.14) for $L = q+1$ is an NND sequence supported on $|k| \le q$, it guarantees NND $\{\hat{c}_k\}$ sequences for estimates of NS coefficients in the form of (2.13). Similarly, by choosing α very close to zero, an NND $\{\hat{c}_k\}$ estimate can be obtained using (2.15). In either case, a severe bias is imposed on the $\hat{c}_k$ estimates. However, since the zeros of $C(z)$ are continuous functions of L or α, one can reduce the bias and still ensure NND estimates. This is accomplished by starting with an NND estimate of $C(z)$, then increasing L or α until the bias is sufficiently reduced or until the estimate becomes negative for some frequencies. Of course, L or α will be data dependent in this case.

3) On the Model for $\hat{R}_k$: In designing a numerator estimator, $\hat{R}_k$ can be an estimate of either (2.7) or (2.8). Two points should be considered. First, if (2.8) is used, the implicit assumption made is that

$$\sum_{i=0}^{p} \hat{a}_i \hat{\rho}(k-i) = 0, \quad k > q. \tag{2.16}$$

Whether or not (2.16) holds depends on the particular estimates used for the AR coefficients and autocorrelations. When (2.16) does hold, $\hat{c}_k$ estimates using (2.8) are equivalent to those using (2.7); however, (2.8)-based estimates require fewer computations to implement. When (2.16) does not hold, the use of (2.8) implicitly incorporates an erroneous assumption. It may be argued that by using (2.7) in $\hat{c}_k$, errors in (2.16) resulting from the AR estimation can be to some extent compensated in the NS estimation; no compensation occurs if (2.8) is used. Thus, we might expect the *spectral* estimates to be more accurate if (2.7) is used (even if the $\hat{c}_k$ parameter estimates are worse due to the compensation effect).

Second, if (2.7) is used for the $\hat{R}_k$, NND estimates of $\{\hat{c}_k\}$ can be obtained (cf., theorem 1). If (2.8) is used, there is no known method of ensuring NND $\{\hat{c}_k\}$ estimates (except in special cases, e.g., when (2.7) and (2.8) are equivalent).

III. A Comparison of Six Numerator Estimators

This section considers six methods for estimating the NS coefficients once the AR coefficients have been estimated [1]–[4], [6], [7]. By fitting each method in the form of (2.9), we can extract some of its properties (such as those discussed in the previous section), and we can modify it to effect certain desired characteristics.

All equations for $\hat{c}_k$ below are for $0 \le k \le q$ only. The $\hat{c}_k$ coefficients for negative k are found by $\hat{c}_{-k} = \hat{c}_k^*$. Moreover, it is assumed that $\hat{c}_k \equiv 0$ for $|k| > q$.

Method 1: The $\hat{c}_k$ coefficients in method 1 are estimated as [1]

$$\hat{c}_k = \sum_{i=0}^{p} \sum_{j=0}^{p} \hat{a}_i \hat{a}_j^* \hat{\rho}_b(k+j-i) \tag{3.1}$$

where $\hat{\rho}_b$ is the standard biased autocorrelation estimator

$$\hat{\rho}_b(k) = \frac{1}{n} \sum_{m=1}^{n-k} x(m+k)\, x^*(m). \tag{3.2}$$

Method 1 directly fits the form (2.9) with $\hat{w}_k = 1$ and with $[\hat{R}_k]_{i,j} = \hat{\rho}_b(k-i+j)$. This method does not guarantee NND estimates; however, $\hat{R}_k$ can be written as

$$\hat{R}_k = \frac{1}{n} X_{1,n+p}^H X_{k,n+p+k} \tag{3.3}$$

so method 1 will give NND estimates if an NND $\{\hat{w}_k\}$ sequence is used. $\hat{R}_k$ makes efficient use of the data as all possible data lags appear in the sum in (3.3). However, $\hat{\rho}_b(k)$ is a biased autocorrelation estimate for $k \ne 0$. If the standard unbiased estimate is used [replacing $1/n$ by $1/(n-k)$ in (3.2)], $\hat{R}_k$ can no longer be written as in (2.10), so NND estimates cannot be guaranteed.

Method 2: Method 2 is proposed in [2]. Here, the $\hat{c}_k$ sequence is found by

$$\hat{c}_k = \sum_{i=0}^{p} \sum_{j=0}^{q-k} \hat{a}_i \hat{a}_j^* \hat{\rho}_b(k-i+j). \tag{3.4}$$

This method is identical to method 1 except that $\hat{R}_k$ is patterned after (2.8) instead of (2.7). Method 2 does not in general guarantee NND estimates even when an NND window is used, because $\hat{R}_k$ cannot be factored in the form $1/JX_{s,t}^H X_{s+k,t+k}$.

Method 3: The estimate for method 3 is proposed in [3], and is given by

$$\hat{c}_k = \sum_{i=0}^{p} [\hat{d}_{i+k}\hat{a}_i^* + \hat{d}_i^*\hat{a}_{i+k}], \quad 0 \le k \le p \tag{3.5}$$

$$\hat{d}_k = \sum_{i=0}^{p} \hat{a}_i r_b^+(k-i), \quad 0 \le k \le p \tag{3.6}$$

$$r_b^+(k) = \begin{cases} 0 & k < 0 \\ r_b(0)/2 & k = 0 \\ r_b(k) & k > 0 \end{cases}. \tag{3.7}$$

By substituting (3.6) and (3.7) into (3.5) and rearranging terms, it can be shown that is a special case of method 2 obtained by setting $q = p$ there (see [8] for details). Thus, all comments concerning method 2 apply here as well.

Method 4: Method 4 was proposed in [4]. First, the forward and backward prediction error sequences are formed; the forward sequence is given by (2.12) and the backward sequence by

$$\hat{b}(k) = \sum_{i=0}^{p} \hat{a}_i x^*(k+i), \quad k = 1, 2, \cdots, n-p. \tag{3.8}$$

The $\hat{c}_k$ estimates are then formed by

$$\hat{c}_k = \frac{\hat{w}_k}{2(n-p)} \left[\sum_{m=p+1}^{n-k} \hat{f}_{m+k}\hat{f}_m^* + \sum_{m=1}^{n-k-p} \hat{b}_{m+k}\hat{b}_m^* \right] \tag{3.9}$$

where $\hat{w}_k$ is some NND weighting function (the full triangular window is suggested in [4]). It is straightforward to show that (3.9) fits in the general estimator form with

$$\hat{R}_k = \frac{1}{2(n-p)} [X_{p+1,n-k}^H X_{p+k+1,n} + Y_{p+1,n-k}^H Y_{p+k+1,n}] \tag{3.10}$$

where $X_{s,t}$ is given by (2.11) and

$$Y_{s,t} = \begin{bmatrix} x(s-p) & x(s-p+1) & \cdots & x(s) \\ x(s-p+1) & x(s-p+2) & \cdots & x(s+1) \\ \vdots & \vdots & & \vdots \\ x(t-p) & x(t-p+1) & \cdots & x(t) \end{bmatrix}. \tag{3.11}$$

If an NND window sequence is used, this estimate is guaranteed to be NND.

Except for the first and last few terms of the two sums in (3.9), they are equivalent to each other. Thus, the few data products which do not appear in both sums are implicitly weighted by one-half of all the other data products. Each element of $\hat{R}_k$ in (3.10) is asymptotically equivalent to (3.3), so method 4 is asymptotically equivalent to method 1. Unlike method 1, however, method 4 can guarantee NND $\hat{c}_k$ estimates while at the same time using unbiased autocorrelation estimates.

Method 5: Method 5 is the smoothed periodogram (SP) method of numerator estimation [5]. This method was recently proposed as an ARMA numerator estimation scheme [6]. The SP method entails first finding the forward and backward prediction error sequences defined in (2.12) and (3.8), and then partitioning them to define the subsequences

$$f_i(k) = \hat{f}[p + k + (i-1)D], \quad 1 \le k \le q+1, \quad 1 \le i \le K \tag{3.12}$$

$$b_i(k) = \hat{b}(k + (i-1)D), \quad 1 \le k \le q+1, \quad 1 \le i \le K \tag{3.13}$$

where D is the shift in time between successive subsequences, and K is the largest integer such that $p + q + 1 + (K-1)D \le n$. The SP numerator estimate is formed by evaluating the periodogram estimates associated with each $\{f_i(k)\}$ and $\{b_i(k)\}$ subsequence, and by then averaging these periodograms. As in method 4, the SP method implicitly forms $\hat{c}_k$ by [8]

$$\hat{c}_k = \frac{1}{q+1} \left\{ \frac{1}{K} \sum_{i=1}^{K} \sum_{m=1}^{q+1-k} f_i(k+m) f_i^*(m) + \frac{1}{K} \sum_{i=1}^{K} \sum_{m=1}^{q+1-k} b_i(k+m) b_i^*(m) \right\}. \tag{3.14}$$

Equation (3.14) is the same as (3.9) when $D = 1$. For $D > 1$, more data lags are included in the (3.9) estimate; there are no data lags between the subsequence in (3.14). This omission of data lags results in a higher asymptotic variance of the $\hat{c}_k$ estimates for method 5 when $D > 1$. Methods 4 and 5 are otherwise identical.

Method 6: Method 6 [7] is a numerator estimator that does not exactly fit the general form of (2.9) but very nearly fits it. In this method, $\hat{c}_k$ can be written as

$$\hat{c}_k = \frac{[\hat{a}^H \hat{R}_{0k} \hat{a}] \cdot [\hat{a}^H \hat{R}_{00} \hat{a}]}{[\hat{a}^H \hat{R}_{kk} \hat{a}]} \tag{3.15}$$

where

$$\hat{R}_{ij} = \frac{1}{n-p} X_{p+1,n-i}^H X_{p+1+j-i,n-j}. \tag{3.16}$$

The second numerator and the denominator terms in (3.15) are identical except that $\hat{R}_{00}$ had k more data products than $\hat{R}_{kk}$. For $n \gg k$, this difference becomes negligible, and these two terms nearly cancel. If the two terms exactly canceled, then (3.15) would be identical to method 4 with only forward prediction errors used there. Since these last two terms cancel asymptotically, method 6 is asymptotically equivalent to method 4 (or to method 1) with no windowing. On the other hand, NND estimates cannot be guaranteed for method 6 because of these two terms that do not exactly cancel for finite n.

IV. Summary

We have presented a general estimator form for the numerator spectral coefficients associated with the ARMA model. This general form is useful in pointing out the asymptotic bias, whether or not NND estimates are ensured, and whether or not an implicit assumption about Yule–Walker equations is made. Six numerator spectrum coefficient estimators were then compared by writing them in this general form. It was shown that

- method 3 is a special case of method 2 with $q = p$;
- method 5 with $D = 1$ is equivalent to method 4;
- method 5 with $D > 1$ is inferior to method 4 because the estimates have larger asymptotic variances;
- methods 2, 3, and 6 cannot in general guarantee NND estimates (methods 1, 4, and 5 can); and
- all six estimates are asymptotically biased only by the (known) window sequence $\{\hat{w}_k\}$ and their variances are asymptotically equal to zero.

References

[1] M. Kaveh, "High resolution spectral estimation for noisy signals," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-27, pp. 286–287, June 1979.

[2] J. F. Kinkel, J. Perl, L. Scharf, and A. Stubberud, "A note on covariance-invariant digital filter design and autoregressive-moving average spectral estimation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-27, pp. 200–202, Apr. 1979.

[3] J. A. Cadzow, "High performance spectral estimation—A new ARMA method," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 524–529, Oct. 1980.

[4] S. Kay, "A new ARMA spectral estimator," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 585–588, Oct. 1980.

[5] P. D. Welch, "The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Trans. Audio Electroacoust.*, vol. AU-15, pp. 70-76, 1967.
[6] J. A. Cadzow, "ARMA spectral estimation: A model equation error procedure," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, Denver, CO, Apr. 9-11, 1980, pp. 598-602.
[7] S. Bruzzone and M. Kaveh, "One some suboptimum ARMA spectral estimators," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 753-755, Dec. 1980.
[8] R. L. Moses, "Design and analysis of fast recursive ARMA spectral estimators," Ph.D. dissertation, Virginia Polytechnic Inst. State Univ., Blacksburg, June 1984.

Modifications to the McClellan, Parks, and Rabiner Computer Program for Designing Higher Order Differentiating FIR Filters

CRAIG A. RAHENKAMP AND B. V. K. VIJAYA KUMAR

Abstract—**Simple modifications to the McClellan, Parks, and Rabiner linear phase finite impulse response (FIR) filter design program are suggested to allow the design of an *n*th-order differentiating FIR filter of arbitrary length for any *n*. Two illustrative examples are also provided.**

I. Introduction

This correspondence presents simple modifications that can be made to the popular McClellan, Parks, and Rabiner FIR filter design Fortran program [1] to extend its capabilities. The algorithm in the original program uses the Remez exchange method [2] to design filters with minimum weighted Chebyshev error in approximating the desired frequency response. The purpose of this correspondence is to document modifications to the program that will allow it to approximate *higher order* differentiating FIR filters in addition to those filter designs which it already supports.

The linear phase FIR filter design program EQFIR, appearing in IEEE collection of digital signal processing programs [1, chapter 5] written by McClellan, Parks, and Rabiner, is used as the basis for this work. The program, as it was originally presented, could not meet our needs [3] for accurately designing 2nd-order through 10th-order FIR differentiators of various lengths. By introducing a new variable into the program to represent the order of the differentiator, and by keeping track of all the symmetries involved, the program presented in [1] was quickly and easily modified to design higher order (order greater than one) differentiating FIR filters.

II. Program Changes

All of the first group of changes, those required to be made to the main routine of the FIR design program, occur within the first 50 executable lines of code. The second group of changes appear in two functions that follow the main program.

Manuscript received March 3, 1986; revised June 3, 1986. This work was supported in part by the Defense Advanced Research Projects Agency and the Air Force Office of Scientific Research under Contract AFOSR-84-0293.

C. A. Rahenkamp was with the Department of Electrical and Computer Engineering, Carnegie-Mellon University, Pittsburgh, PA 15213. He is now with the IBM Federal Systems Division, Manassas, VA 22110.

B. V. K. Vijaya Kumar is with the Department of Electrical and Computer Engineering, Carnegie-Mellon University, Pittsburgh, PA 15213.

IEEE Log Number 8610379.

One new parameter is required in the program to represent the order of the differentiator—IORD. When designing FIR filters other than differentiators with this program, IORD is set to 1. By replacing the first READ statement in the program with the following two lines:

```
READ (INPUT,110) NFILT,JTYPE,NBANDS,LGRID,IORD
IF(IORD.LE.0) IORD=1
```

the IORD parameter is correctly entered. As a consequence, the following FORMAT statement in the original program:

```
110 FORMAT(4I5)
```

must be replaced by the following statement:

```
110 FORMAT(5I5).
```

The symmetries of the filter must next be properly established. This requires modification of the parameter NEG. NEG is assumed to be zero for any even symmetric filter (e.g., a multipass-band/stop-band filter) and 1 for any odd symmetric filter. This requires that NEG = 0 for any differentiator of even-order and NEG = 1 for any odd-order differentiator. Following the lines below which presently set NEG,

```
NEG=1
IF(JTYPE.EQ.1) NEG=0
```

the following lines must be inserted:

```
IF(JTYPE.NE.2)GOTO 126
NEG=IORD/2
NEG=IORD-2*NEG
```

and the line that follows modified by a jump label:

```
126 NODD=NFILT/2
```

To prevent divide by zero errors, the variable DELF must be properly initialized. This is accomplished by modifying the line that reads

```
IF(NEG.EQ.0) GO TO 135
```

in the portion of the code that sets up the dense grid to read as below:

```
IF(JTYPE.EQ.1) GO TO 135
```

The following two changes must be made in the section that calculates the desired magnitude response so that the variable IORD is passed to the magnitude response function. The line

```
DES(J)=EFF(TEMP,FX,WTX,LBAND,JTYPE)
```

must be changed to

```
DES(J)=EFF(TEMP,FX,WTX,LBAND,JTYPE,IORD)
```

and the line

```
DES(J-1)=EFF(FUP,FX,WTX,LBAND,JTYPE)
```

must be changed to the following:

```
DES(J-1)=EFF(FUP,FX,WTX,LBAND,JTYPE,IORD)
```

Finally, a modification is made to the output section of the main routine. This modification occurs much further down in the code—close to the end of the main routine. The change is made to allow the output to reflect the input parameter IORD. By modifying the lines that read