COLORADO STATE
UNIVERSITY
FORT COLLINS COLORADO
80523

department of electrical engineering

AD A 039970

(1)

See 1473 in back

Statistical Design of Autoregressive-Moving Average Digital Filters

by

Louis L. Scharf

James C. Luby

DDC
RECEIVED
MAY 27 1977
C

ONR Technical Report #21

May 1977

Prepared for the Office of Naval Research
under Contract N00014-75-C-0518
with joint sponsorship of NAVALEX 320

L. L. Scharf and M. M. Siddiqui, Principal Investigators

Reproduction in whole or in part is permitted
for any purpose of the United States Government

Approved for public release; distribution unlimited

D No. ____
DDC FILE COPY

Statistical Design of Autoregressive-Moving Average Digital Filters

by

Louis L. Scharf

James C. Luby

ONR Technical Report #21

April 1977

Prepared for the Office of Naval Research
under Contract N00014-75-C-0518
with joint sponsorship of NAVALEX 320

D D C
RECEIVED
MAY 27 1977
C

L. L. Scharf and M. M. Siddiqui, Principal Investigators

Reproduction in whole or in part is permitted
for any purpose of the United States Government

Approved for public release; distribution unlimited

Statistical Design of Autoregressive-Moving Average

Digital Filters[1]

ACCESSION for
NTIS White Section ☑
DDC Buff Section ☐
UNANNOUNCED ☐
JUSTIFICATION.....
BY.....
DISTRIBUTION/AVAILABILITY CODES
Dist. AVAIL. and/or SPECIAL
A

by

Louis L. Scharf[2]

James C. Luby[2]

Submitted to

IEEE Transactions on Acoustics, Speech, and Signal Processing

May 1977

[1]This work has been supported by the Office of Naval Research, Statistics and Probability Branch, Arlington, VA, under Contract N00014-75-C-0518, with the joint sponsorship of the Naval Electronic Systems Command, Code 320.

[2]Electrical Engineering Department, Colorado State University, Fort Collins, Colorado 80523

Abstract

I. Introduction

II. Beginning the Design

III. Designing an Autoregressive Approximation

IV. Obtaining Unit-Pulse and Covariance Sequences from the AR Design

V. Designing with the Mullis-Roberts Algorithm

VI. Systematic Design Steps

VII. Designs

VIII. Conclusions

References

Abstract

Procedures are presented for the systematic design of digital filters that contain poles and zeros. The procedures are simple, fast, and effective. All of the important algorithms are of the Levinson-type. The first key idea in the paper is that one may begin a design by posing a linear prediction problem for a stochastic sequence. The second is that a high-order "whitening" filter may be constructed for this sequence and "inverted" to yield a high-order all-pole filter whose spectrum approximates the spectrum of the stochastic sequence. The third key idea is that the all-pole filter may be used to generate consistent unit pulse and covariance sequences for use in the Mullis-Roberts algorithm. This algorithm is then used to obtain a low-order digital filter, with poles and zeros, that approximates the high-order all-pole filter. The results demonstrate that the Mullis-Roberts algorithm, together with the design philosophy of this paper, may be used with profit to reduce filter complexity and to design spectrum-matching digital filters.

I. Introduction

This is a paper about the systematic design of autoregressive-moving average (ARMA) digital filters. There are three key ideas in the paper. The first is that one may begin the design of an ARMA digital filter by specifying the power spectral density of a stochastic sequence and then posing a classical linear prediction problem for the stochastic sequence. The second is that this prediction problem may be solved by designing a high-order moving average (MA) prediction filter. This prediction filter is related to an MA whitening filter which may, in turn, be "inverted" to give an autoregressive (AR) filter termed an inverse filter. Identical argumentation lies at the heart of AR or maximum entropy (ME) spectrum analysis and explains why the magnitude-squared frequency response of the AR filter should approximate the power spectrum of the stochastic sequence [1]. The appropriate design equations are linear equations (termed normal equations) that may be efficiently solved using Levinson-type algorithms. For our purposes the value of the high-order AR filter is simply that it provides a handy mechanism for generating consistent unit pulse and covariance sequences $\{h_k\}_0^\infty$ and $\{r_k\}_{-\infty}^\infty$, respectively. These sequences approximate the unit pulse and covariance sequences of an idealized, unrealizable, digital filter.

The third key idea is that the sequences $\{h_k\}_0^\infty$ and $\{r_k\}_{-\infty}^\infty$ may be used in the approximation algorithm published by Mullis and Roberts in their remarkable paper on the use of first- and second-order information in discrete-time system design [2]. From $\{h_k\}_0^\infty$ and $\{r_k\}_{-\infty}^\infty$ for the high-order AR, the finite-length sequences $\{h_k\}_0^M$ and $\{r_k\}_0^N$ are used in the Mullis-Roberts algorithm to design an ARMA (N,M) digital filter. The notation ARMA (N,M) denotes a digital filter with N poles and M zeros. Correspond-

ingly, AR (N_1) will denote an all-pole digital filter with N_1 poles and MA (M_1) will denote an all-zero digital filter with M_1 zeros.

Our results indicate that the design procedure outlined above is simple, effective, and fast. All of the important algorithms are Levinson-type algorithms. Moderate-order filters (e.g., N=16 and M=16) may be designed in approximately 8 seconds of CDC 6400 CPU time. Design parameters are center frequency, bandwidth, and stopband rejection. Even for low-order filters (e.g. N=8 and M=8) one may achieve 60 dB rejection over a transition band that is a small percentage of the foldover frequency with approximately 3dB passband ripple.

As an approximation tool the Mullis-Roberts algorithm is very effective: for example, an ARMA (N,M) filter of complexity C = N + M is shown in many instances to be a very good approximation to an AR (N_1) filter for $C \ll N_1$. Designs for $N_1 = 256$ and C = 32, and N = 64 and C = 16 illustrate the point. Our experience indicates that the designs presented here are superior to those achieved by simply matching the impulse sequence $\{h_k\}_0^{N+M}$ using the Burrus-Parks algorithm [3]. Furthermore there is no difficulty with stability because both the AR (N_1) and ARMA (N,M) filters are guaranteed to be stable.

Preliminary design efforts along the lines of this paper were reported in [4]. The designs were begun with a moderate-order MA (M_1) filter (e.g., $M_1 = 32$) designed using the methodology of [5]. The MA filter was used to generate consistent unit-pulse and covariance sequences for use in the Mullis-Roberts algorithm. The results reported were mixed because the unit-pulse and covariance sequences were not well-suited for subsequent approximation using the Mullis-Roberts algorithm. In our opinion, and the results seem to corroborate this view, one should begin ARMA (N,M)

approximations with very high-order AR (N_1) designs (e.g., $N_1 = 256$) rather than very long or short MA designs. The reasons are three-fold: (i) AR power spectrum approximations are well-understood [6] and virtually free in terms of numerical design effort, (ii) the sequences $\{h_k\}_0^\infty$ and $\{r_k\}_{-\infty}^\infty$ are obtained from simple regression equations, and (iii) the AR sequences $\{h_k\}_0^\infty$ and $\{r_k\}_{-\infty}^\infty$ are infinite-length and, therefore, apparently better suited to subsequent matching and approximation by infinite length ARMA unit-pulse and covariance sequences.

We remark that the approximation algorithm of Mullis and Roberts gives exact matching of the ARMA (N,M) and AR (N_1) unit pulse sequences over M + 1 indices and only approximate matching of the covariance sequences over N + 1 indices. When only a covariance sequence is to be matched, then the Mullis-Roberts algorithm reduces to the Levinson algorithm for obtaining an exact covariance match over N + 1 indices.

We feel the statistically-related filter design procedures reported here and in [5] and [7] are becoming well-enough understood, and are yielding attractive-enough designs, to warrant serious attention from designers. One needs only to view the digital filter design problem as a problem of statistically designing a linear minimum mean-squared error filter or predictor. Elegant solutions abound. An added dividend paid by this way of thought is that a class of digital filter designs becomes a logical branch of linear minimum mean-squared error filtering theory. Some find this comforting. Others don't need it.

II. Beginning the Design

The design begins with specification of a bandlimited power spectrum $G(f)$ of the type illustrated in Figure 1a. It may be helpful to visualize the spectrum on the passband interval $-W < f < W$ as a signal spectrum $S(f)$ and the spectrum on the remainder of the frequency interval (the stopband) as a noise spectrum $N(f)$. See [5]. Then $S(f)$ corresponds to the desired passband characteristic and $N(f)$ corresponds to the desired stopband characteristic. Center frequency f_0 ($f_0 = 0$ in Figure 1), signal bandwidth W, stopband rejection η, and total bandwidth $1/2T$ are design parameters. The rejection η may be visualized as a signal-to-noise ratio $\eta = A_s/A_n$. In the design of MA filters it has been found useful to include a transition band [5]. In the design of ARMA filters beginning with AR filters we have not discovered a useful way to specify a transition band.

The next step is to periodically extend the spectrum $G(f)$ using period $1/T$ (or foldover frequency $1/2T$) and scale it by $\frac{1}{T}$ to obtain a periodic spectrum $G_p(f)$:

$$G_p(f) = \begin{cases} \frac{1}{T}G(f) \quad , & |f| < 1/2T \\ \frac{1}{T}G(f - \frac{m}{T}), & m/T - 1/2T < f < m/T + 1/2T \quad , \quad m = 0, \pm 1, \pm 2, \ldots \end{cases} \tag{1}$$

This is illustrated in Fig. 1b. The Fourier coefficients

$$c_k = T \int_{-1/2T}^{1/2T} G_p(f) e^{-j2\pi fkT} df \quad , \quad k = 0, \pm 1, \pm 2, \ldots \tag{2}$$

specify a covariance sequence $\{c_k\}_{-\infty}^{\infty}$ with $c_k = c_{-k}$. The sequence $\{c_k\}_{-\infty}^{\infty}$ may be thought of as the covariance sequence of a zero-mean sampled data sequence $\{x(kT)\}_{-\infty}^{\infty}$ obtained by sampling a continuous-time random process $\{x(t)\}_{-\infty}^{\infty}$ that has bandlimited power spectral density $G(f)$. Then,

of course, $c_k = c(\tau=kT)$ where

$$c(\tau) = \int_{-1/2T}^{1/2T} G(f)e^{-j2\pi f\tau}df \tag{3}$$

is the covariance of the process $\{x(t)\}_{-\infty}^{\infty}$.

Another connection between $\{c_k\}$ and the periodically extended spectrum $G_p(f)$ is

$$G_p(f) = M(z)M(z^{-1})\big|_{z=\exp(j2\pi fT)}$$

$$M(z)M(z^{-1}) = \sum_{k=-\infty}^{\infty} c_k z^{-k} \tag{4}$$

This says $\{c_k\}_{-\infty}^{\infty}$ is the covariance of the output of a digital filter $M(z)$ that is driven by a white sequence with unit variance and that the periodically-extended spectrum $G_p(f)$ is the magnitude-squared frequency response of the digital filter $M(z)$.

The periodic spectrum $G_p(f)$ illustrated in Figure 1 is a legitimate discrete-time spectrum. However, it is not rational and is, therefore, not the spectrum of an autoregressive moving average digital filter. This is another way of saying the $M(z)$ of (4) is not a transfer function for a filter that can be realized using a finite number of multipliers, adders, and delays (memory). Thus the approximation problem is one of designing an ARMA filter $H(z)$ whose rational spectrum $H(z=e^{j2\pi fT})\cdot H(z^{-1}=e^{j2\pi fT})$ approximates the irrational spectrum $G_p(f)$. Of course, $H(z=e^{j2\pi fT})H(z^{-1}=e^{j2\pi fT})$ is simply the magnitude-squared frequency response of the filter $H(z)$. This brings us to the next step in our design procedure.

III. Designing an Autoregressive Approximation

The problem now is the following: given a covariance sequence $\{c_k\}_{-\infty}^{\infty}$ corresponding to the desired spectrum $G_p(f)$, find a realizable filter $H(z)$ whose periodic spectrum well-approximates $G_p(f)$. Call $\{r_k\}_{-\infty}^{\infty}$ the covariance sequence of the filter $H(z)$. The following relationships are in force:

$$H(z)H(z^{-1}) = \sum_{k=-\infty}^{\infty} r_k z^{-k}$$

$$r_k = \frac{1}{2\pi j} \oint_C H(z)H(z^{-1})z^{k-1}dz \tag{5}$$

$$H(z=e^{j2\pi fT})H(z^{-1}=e^{j2\pi fT}) = \sum_{k=-\infty}^{\infty} r_k \, e^{-j2\pi fkT}$$

It is clear that for purposes of matching spectra, the covariances come into play in a more fundamental way than do the unit pulse sequences. In (5) the contour C lies within the annulus of uniform convergence of $H(z)H(z^{-1})$.

One approach to the approximation problem is to design an AR (N_1) filter

$$H(z) = \frac{\kappa}{1 - \sum_{\ell=1}^{N_1} \alpha_\ell z^{-\ell}} \tag{6}$$

in such a way that

$$r_k = c_k \quad , \quad k = 0, \pm 1, \ldots \pm N_1 \tag{7}$$

For large values of N_1, the matching of $\{r_k\}_0^{N_1}$ and $\{c_k\}_0^{N_1}$ provides effective approximation of $G_p(f)$ by $H(z=e^{j2\pi fT})H(z^{-1}=e^{j2\pi fT})$. See equations (2) and (5). The result of (7) is achieved with the filter of (6) by solving the normal equations

$$C\bar{\alpha} = \bar{c}$$

$$C = \begin{matrix} c_0 & c_1 & \cdots & c_{N_1-1} \\ c_1 & c_0 & & \\ \vdots & & & \\ c_{N_1-1} & \cdots & & c_0 \end{matrix} \tag{8}$$

$$\bar{c}' = (c_1, c_2, \ldots, c_{N_1})$$

$$\bar{\alpha}' = (\alpha_1, \alpha_2, \ldots, \alpha_{N_1})$$

$$\kappa^2 = c_0 - \sum_{\ell=1}^{N_1} \alpha_\ell c_\ell$$

Here the prime denotes matrix transpose.

These normal equations arise over and over again in speech processing and autoregressive spectrum analysis. When statistically fluctuating data enters the picture these equations are replaced by the famous Yule-Walker equations involving estimates of c_k.

Due to the Toeplitz nature of C and the relation $\bar{c}$ bears to C, the linear equation $C\bar{\alpha} = \bar{c}$ may be solved very efficiently using a Levinson-type algorithm. In the FORTRAN program of Appendix I, the subroutine TPLSLV implements such an algorithm. See [8] for a listing of the algorithm and a discussion of computational demands.

In summary, the AR (N_1) coefficients $\{\alpha_\ell\}_1^{N_1}$ are obtained by solving (8) with the $\{c_k\}_0^{N_1}$ obtained from the desired spectrum $G_p(f)$ according to (2). For all of our designs the c_k may be obtained analytically as follows:

$$C_k = A_s 2TW \operatorname{sinc}[2\pi WkT] + A_n[1-2TW]\cos[(2TW+1)\tfrac{\pi}{2}k]\operatorname{sinc}[(1-2TW)]\tfrac{\pi}{2}k] \tag{9}$$

Figure 2 shows the results of several AR designs. Each design was begun with a $G_p(f)$ of the form illustrated in Fig. 1. The idealized $G_p(f)$ is superimposed on each design. Important parameters are included in the figures.

There is one other interpretation of the H(z) given in (6) that is worth noting, even though it has been noted many places. The filter

$$L(z) = \sum_{\ell=1}^{N_1} \alpha_\ell z^{-\ell} \tag{10}$$

with the $\{\alpha_\ell\}$ given by (8) is the linear MA filter of order N_1-1 that minimizes the mean squared prediction error

$$\xi^2 = E[x(mT) - \sum_{\ell=1}^{N_1} \alpha_\ell \, x((m-\ell)T)]^2 \tag{11}$$

Here $\{x(\ell T)\}_{-\infty}^{\infty}$ is the sampled-data sequence that has spectrum $G_p(f)$. The filter

$$K(z) = 1 - \sum_{\ell=1}^{N_1} \alpha_\ell z^{-\ell} \tag{12}$$

is the corresponding "whitening" filter. Of course $H(z) = 1/K(z)$. Therefore, to the extent that K(z) whitens $G_p(f)$, H(z) inherits the spectral properties of $G_p(f)$. It is well-known that an H(z) designed as outlined above is stable [9].

IV. Obtaining Unit-Pulse and Covariance Sequences from the AR Design

No one wants a high-order AR filter, regardless of its frequency response characteristics. However, the high-order AR design serves another very useful purpose: it provides a ready-made generator for a covariance sequence $\{r_k\}_{-\infty}^{\infty}$ and a causal unit pulse sequence $\{h_k\}_0^{\infty}$ that are consistent in the sense that

$$\begin{aligned} r_k &= \sum_{\ell=0}^{\infty} h_\ell h_{\ell+k} \quad , \qquad k = 0, 1, 2, \ldots \\ r_{-k} &= r_k \end{aligned} \tag{13}$$

The appropriate equations for the generation of the unit pulse sequence $\{h_k\}_0^{\infty}$ are

$$h_k = \begin{cases} 0 \quad , & k < 0 \\ \kappa\delta_k + \sum_{\ell=1}^{N_1} \alpha_\ell \, h_{k-\ell} \quad , & k \geq 0 \end{cases} \tag{14}$$

Here the coefficients $\{\alpha_\ell\}_1^N$ are the feedback coefficients of H(z). See equations (6) - (8). The symbol δ_k denotes the Kronecker delta.

The generating equations for $\{r_k\}_{-\infty}^{\infty}$ are not much more difficult. The following linear relationship holds:

$$\begin{aligned} r_k &= \sum_{\ell=1}^{N_1} \alpha_\ell \, r_{k-\ell} \\ r_{-k} &= r_k \end{aligned} \tag{15}$$

So, given the N_1 covariances $\{r_\ell\}_1^{N_1}$, for example, one may solve for r_{N_1+1}, and so on. Of course, the $\{r_\ell\}_1^{N_1}$ are available as $r_\ell = c_\ell$, $\ell = 0, 1, \ldots, N_1$, from the AR design procedure of Section III. For

completeness we show in Appendix I how to solve for $\{r_\ell\}_0^{N_1}$ in terms only of the $\{\alpha_\ell\}_1^{N_1}$, and thereby completely characterize the covariance sequence $\{r_\ell\}_{-\infty}^{\infty}$. We re-iterate that the calculation of Appendix II is <u>not</u> required in our design procedure because the $\{r_\ell\}_0^{N_1}$ are available as $r_\ell = c_\ell$, $\ell = 0, 1, \ldots, N_1$ with the c_ℓ given by (2).

Here is where we stand: we have at our disposal a systematic technique for generating a high-order AR (N_1) filter that "approximates" the idealized spectrum $G_p(f)$. This AR (N_1) filter is characterized by its impulse and covariance sequences $\{h_k\}_0^{\infty}$ and $\{r_k\}_{-\infty}^{\infty}$ which are easily obtained as outlined above. The sense of the approximation is that $r_k = c_k$, $k = 0, 1, 2, \ldots, N_1$.

In the following section we use the algorithm of Mullis and Roberts to approximate this generally high-order AR (N_1) filter with a much lower order ARMA (N,M) filter.

V. Designing with the Mullis-Roberts Algorithm

Call H(z) the transfer function of our AR (N_1) design. Let $\hat{H}(z)$ be the transfer function of an ARMA (N,M) filter:

$$\hat{H}(z) = \frac{Q(z)}{A(z)}$$

$$Q(z) = q_0 + q_1 z^{-1} + \ldots + q_M z^{-M} \tag{16}$$

$$A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_N z^{-N}$$

The problem studied by Mullis and Roberts is one of minimizing the approximation error

$$\varepsilon^2 = T \int_{-1/2T}^{1/2T} |A(z=e^{j2\pi fT})|^2 |H(z=e^{j2\pi fT}) - \hat{H}(z=e^{j2\pi fT})|^2 df \tag{17}$$

In the general case H(z) is arbitrary. For our purposes it will always be the transfer function of a high-order AR (N_1) filter. The error ε^2 may be written

$$\varepsilon^2 = T \int_{-1/2T}^{1/2T} |H(z=e^{j2\pi fT})A(z=e^{j2\pi fT}) - Q(z=e^{j2\pi fT})|^2 df \tag{18}$$

The main virtue of this error measure is that it leads to a tractable minimization problem [2]. It is a straight-forward exercise to define $\Lambda(z) = H(z)A(z) - Q(z)$, invoke Parseval's Theorem, and use (13) and (18) to write

$$\varepsilon^2 = \sum_{\ell=0}^{N} a_\ell \sum_{m=0}^{N} a_m r_{|\ell-m|} - 2 \sum_{k=0}^{M} q_k \sum_{\ell=0}^{N} a_\ell h_{k-\ell} + \sum_{k=0}^{M} q_k^2 \tag{19}$$

The minimization problem is one of minimizing ε^2 with respect to the ARMA parameters $\{q_\ell\}_0^M$ and $\{a_\ell\}_0^N$, subject to the constraint that $a_0 = 1$. Write this as

$$\min_{\bar{a},\bar{q}} [\varepsilon^2 - 2\lambda(\bar{a}'\bar{\psi} - 1)] \tag{20}$$

where λ is a Lagrange multiplier and

$$\begin{aligned} \bar{a} &= (a_N, a_{N-1}, \ldots, a_0)' \\ \bar{q} &= (q_M, q_{M-1}, \ldots, q_0)' \\ \bar{\psi} &= (0, 0, \ldots, 1) \end{aligned} \tag{21}$$

Note $\bar{a}'\bar{\psi} = a_0$.

First minimize with respect to $\bar{q}$. The constraint plays no role and the result is

$$q_k = \sum_{\ell=0}^{N} a_\ell h_{k-\ell} \quad , \quad k = 0, 1, \ldots, M \tag{22}$$

with the $\{h_k\}$ corresponding to H(z) and the $\{a_\ell\}_0^N$ yet to be determined.

Substitute this result into (19) to obtain

$$\varepsilon^2 = \bar{a}'K\bar{a}$$
$$K = \{r_{|\ell-m|} - \sum_{k=0}^{M} h_{k-\ell}h_{k-m}\} \tag{23}$$

Here the notation $K = \{d_{\ell,m}\}$ denotes that $d_{\ell,m}$ is the (ℓ,m)-th term of the $(N+1) \times (N+1)$ matrix K. The matrix K is positive semi-definite if and only if $\{r_k\}_{-\infty}^{\infty}$ and $\{h_k\}_0^{\infty}$ are consistent, as they are by virtue of our construction method.

Now our constrained minimization problem yields the following solution for $\bar{a}$:

$$K\bar{a}^* = \lambda\bar{\psi}$$
$$\bar{a}^* = \lambda K^{-1}\bar{\psi} \tag{24}$$

Invoke the constraint to get

$$\bar{a}^* = (k_N/k_0,\ k_{N-1}/k_0,\ \ldots,\ 1) \tag{25}$$

where $\bar{k} = (k_N, \ldots, k_0)$ is the last column of K^{-1}. Substitution of $\bar{a}^*$ into (22) yields $\bar{q}^*$ and completes the design of the ARMA (N,M) filter

$$H(z) = \frac{q_0^* + q_1^* z^{-1} + \ldots + q_M^* z^{-M}}{1 + a_0^* z^{-1} + \ldots + a_N^* z^{-N}} \tag{26}$$

Call $\{\hat{h}_k\}_0^{\infty}$ the unit pulse sequence and $\{\hat{r}_k\}_{-\infty}^{\infty}$ the covariance sequence corresponding to $\hat{H}(z)$. Since $\bar{a}^*$ is simply the solution to a normal equation involving positive-definite K, it follows that $\{\hat{h}_k\}$ is absolutely summable. That is $\hat{H}(z)$ is stable. Further, as shown by Mullis and Roberts, $\hat{h}_k = h_k$, $k = 0, 1, \ldots, M$. The covariances $\hat{r}_k$ and r_k are not equal.

All of the results of this section are contained in [2].

VI. Systematic Design Steps

The design steps are summarized below and illustrated in Figure 3.

1. Specify a periodic power spectrum $G_p(f)$.
2. Invert $G_p(f)$ to get a design covariance sequence $\{c_k\}_0^{N_1}$.
3. Solve the normal equations $C\bar{\alpha} = \bar{c}$ to obtain an AR (N_1) approximation. Typically choose N_1 large (e.g. $N_1 = 256$).
4. Generate $\{r_k\}_0^{N_1} = \{c_k\}_0^{N_1}, N_1 \leq N$; generate $\{h_k\}_0^M$ using the AR generating equation (14).
5. Construct the matrix K.
6. Solve for $\bar{a}^*$ and $\bar{q}^*$.
7. Scale $\hat{H}(z)$ to get desired dc response $H_0(z=1) = H_0$.

Actually, steps 5 and 6 are replaced by a Levinson-type algorithm presented by Mullis and Roberts for efficiently obtaining $\bar{a}^*$ and $\bar{q}^*$.

The FORTRAN listing of Appendix II describes a program written for a CDC 6400 computer with SCOPE compiler to implement the design steps above. The program was used to generate all of the designs that follow in Section VII. The number of storage locations required for the TPLSV subroutine to solve (8) is $3N_1$. The number of multiplies is approximately $2N_1^2$ and the number of adds is approximately N_1. The number of storage locations required to solve for $\bar{a}^*$ using the Mullis-Roberts algorithm is O(N). The number of multiplies is $O(N^2)$. The notation O(N) indicates here that O(N)/N is a constant.

VII. Designs

The designs presented in Figures 4-8 all have a design cutoff frequency of 0.2Hz and a foldover frequency of 1.0Hz. This does not mean the 3dB point, for example, is under careful control. It only means the ideal spectrum with which the design was begun had a cutoff frequency of 0.2Hz. Our convention in the figures is that ARMA(N,M) always denotes a filter designed according to the procedure of Section VI. Of course, the Chebyshev and Butterworth design of Figures 7 and 8 are also autoregressive moving average filters. These filters are denoted CH(N,N) and BW(N,N), respectively.

The results of Figure 4 illustrate how low-order ARMA filters may be used to approximate high-order AR filters. Figure 4(a) illustrates that an ARMA (8,8) filter is a very effective approximation to an AR(64) filter when only 30dB stopband rejection is desired. The complexity of the ARMA (8,8) is one-fourth the complexity of the AR(64) and the passband ripple of the ARMA (8,8) is very low. Figure 4(b) illustrates that an ARMA (32,32) filter is an effective approximation to an AR(256) when 90dB rejection is desired. Figure 5(a) illustrates what happens when one fixes the number of poles (N=8) and the number of zeros (M=4) in a relatively low complexity design and then increases the stopband rejection. At 30dB rejection the filter characteristic is smooth in the stopband and cuts off sharply. At 90dB rejection the ripple in the passband is severe. Figure 5(b) illustrates that if additional zeros are added to increase the complexity, control is maintained over the passband ripple even as the stopband rejection is increased to 90dB.

Figure 6 shows that when complexity is already high (N=16, $M \geq 8$) and the stopband rejection moderate (60dB), the addition of zeros does not

dramatically influence the magnitude-squared response.

Figure 7 shows ARMA (8,M) approximations to ideal lowpass spectra and comparisons with Butterworth and Chebyshev designs of order 8 and complexity 16. If only 30dB stopband rejection is desired then an ARMA (8,4) filter can be designed that yields smoother passband characteristics than the Cheybshev filter and sharper cutoff than either the Chebyshev filter or the Butterworth filter. Of course, the passband ripple is larger than that of the Butterworth filter. The ARMA (8,4) filter is less complex than the Chebyshev and Butterworth filters by a factor of 3/4. As rejection is increased to 60dB, the passband ripple of an ARMA (8,8) increases, but remains smaller than the Chebyshev ripple everywhere in the passband except near cutoff. The cutoff is comparable.

Figure 8 shows ARMA (16,M) approximations to ideal lowpass spectra and comparisons with Butterworth and Chebyshev designs of order 16 and complexity 32. If only 60dB stopband rejection is required, then an ARMA (16,8) provides cutoff comparable to that of the Chebyshev with smaller ripple except near cutoff. For 90dB rejection an ARMA (16,16) provides cutoff comparable to the Chebyshev filter of the same complexity with less passband ripple except near cutoff.

In all of our designs run to date, well-conceived ARMA designs exhibit relatively little passband ripple, except very near the cutoff frequency, and cutoff characteristics comparable to Chebyshev characteristics. When filter complexity is high and stopband rejection low, then the filter characteristics can approximate ideal lowpass characteristics much more closely than equivalent-complexity Butterworth or Chebyshev designs. Reference the ARMA (8,8) filter of Figure 4(a), the ARMA (8,12) - 30dB filter of Figure 5(a), and the ARMA(16,24) filter of Figure 6.

VIII. Conclusions

The results presented here are only representative of what one may achieve. Extensions to bandpass, high-pass, band reject, and comb filters are straightforward. Several points should be made: (i) there is no requirement in these designs to begin with a rational analog filter; (ii) there is nothing fundamental about the shape of the G(f) we have started with - an inventive designer may experiment with his own choices; (iii) the procedures presented here permit the designer to exercise implicit control over center frequency, bandwidth, and stopband rejection; and (iv) one may systematically design filters with arbitrary numbers of poles and zeros.

The results of Section VII do not say the statistical designs of this paper are better or worse than more classical designs. Only different. The new designs do, however, offer much more flexibility.

We speculate that the fitting of long AR models and subsequent approximation with the Mullis-Roberts algorithm may provide a useful method of fitting ARMA spectral models to random data. The question of order determination would, of course, be crucial here as it is in AR model-fitting [6].

APPENDIX I: Generating a Covariance Sequence from an AR Model

Write out (15) for $k = 1, 2, \ldots, N_1$ and order the equations in the following way:

$$Q\bar{r} = \bar{r}$$

$$Q = \begin{bmatrix} 0 & \alpha_1 & \alpha_2 & \alpha_3 & \cdots & \alpha_{N_1} & \\ \alpha_1 & \alpha_2 & \alpha_3 & \cdots & \alpha_{N_1} & 0 & \\ \alpha_2 & (\alpha_1+\alpha_3) & \alpha_4 & \alpha_5 \cdots & \alpha_{N_1} & 0 & 0 \\ \cdot & & & & & & \\ \cdot & & & & & & \\ \cdot & & & & & & \\ \alpha_{N_1} & \alpha_{N_1-1} & \cdot & \cdot & \cdot & \alpha_1 & 0 \end{bmatrix}$$

$$\bar{r} = (r_0, r_1, \ldots, r_{N_1})$$

The matrix Q is generated as follows: begin with the first row $(0, \alpha_1, \alpha_2, \ldots \alpha_{N_1})$; left-shift this row (n-1) times and add the mth overflow to the (n,m+1) term to get the nth row. For example, the fourth row is generated in the following way:

$$\begin{array}{cccccccccc} & & & 0 & \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 \cdots & \alpha_{N_1} \\ 0 & \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 & & & & \\ & & & \alpha_3 & \alpha_2+\alpha_4 & \alpha_1+\alpha_5 & \alpha_6 & \cdots & \alpha_{N_1} & 0 \end{array}$$

The vector of covariances $\bar{r}$ is obtained by solving the eigenvalue problem

$$Q\bar{\ell} = \lambda\bar{\ell}$$

for the eigenvector $\bar{\ell}$ $(= \bar{r})$ corresponding to $\lambda = 1$.

Appendix II: FORTRAN Program for Design of ARMA Filters

```
      PROGRAM ARFILT(INPUT,OUTPUT,FILMPL,TAPE5=INPUT,TAPE6=OUTPUT)
C        PROGRAM FOR STATISTICAL DESIGN OF AR AND ARMA DIGITAL FILTERS
C        WRITTEN BY J. LUBY, COLORADO STATE UNIV., WINTER, 1976-77.
C        PROGRAM SET UP FOR LOWPASS DESIGNS AT PRESENT(AR OR ARMA).
C        DATA READ IN SETS OF 6 CARDS.  VARIABLE NOC TELLS PROGRAM
C        THE NUMBER OF 6 CARD SETS TO READ IN.(SEE FIRST READ STATEMENT)
C
      INTEGER BS
      INTEGER NMR(10),NEN(10)
      INTEGER TOOL(10)
      DIMENSION HQ(257)
      DIMENSION E(257)
      DIMENSION H(257),A(257),Q(257)
      DIMENSION QN(257)
      DIMENSION PN(257),RS(257),R(257),PP(257)
      DIMENSION WAR(257)
      DIMENSION PHAR(512),HAR(512),WSQ(512),PHASE(512),FREQ(512)
C
C        ALL DATA CARDS MUST BE PRESENT EVEN IF THEY ARE
C        NOT APPLICABLE TO THE PRESENT RUN.
C        BLANK CARDS MAY BE INSERTED FOR N/A DATA CARDS.
C        (FOR AR DESIGNS, A BLANK CARD IS INSERTED FOR THE
C        NUMERATOR ORDER CARD(CARD 5 IN THE SET OF 6))
C        NOTE...EACH DATA SET ALLOWS DESIGN OF AN ARMA OR AN AR,
C        BUT NOT BOTH.
C
C        NOC=NUMBER OF DATA SETS
C        NOC  .LE. 10
      READ (5,250) NOC
C
      DO 240 IX = 1,NOC
C
C     CARD1
C        OPTION CARD
C        THIS CARD IS READ IN 10I1 FORMAT
C        A 1 OR 0 IS PUNCHED IN COLUMNS 1-8, WITH THE FOLLOWING MEANING-
C        1=DESIGN AR FILTER                0=NO AR DESIGN
C        1=PLOT AR PHASE                   0=NO PLOT
C        1=DUMP RELEVENT COEFS.            0=NO DUMP
C        1=DESIGN ARMA FILTER              0=NO ARMA DESIGN
C        1=PLOT ARMA PHASE                 0=NO PLOT
C        1=DUMP ARMA COEF.                 0=NO DUMP
C        1=PLOT MAG. SQUARED               0=NO MAG. SQ. PLOT
C        1=PLOT ON MICROFILM               0=NO PLOT
C
C        ALL COEFFICIENTS IN DUMP ARE NORMALIZED FOR 0 D.C. ON PLOT.
C
      READ (5,270) (TOOL(I),I = 1,10)
C
C     CARD2
C        JJA=NUMBER OF DIFFERENT FILTER ORDERS TO BE DESIGNED
C        UNDER THE ACTIVE OPTION CARD.
C        JJA.LE.9
C        NOTE THAT JJA IS THE DO LOOP PARAMETER IN THE READ
C        STATEMENTS OF CARDS 5 AND 6.
C        FORMAT 1I1
      READ (5,250) JJA
C
C     CARD3
C        BWS=SIGNAL BANDWIDTH IN HZ          BWN=NOISE BW IN HZ
C        BWS+BWN MUST BE .LE. FOLDOVER FREQ.
C        FORMAT  3F3.2
      READ (5,280) BWS,BWN
C
```

```
C     CARD4
C          IONE IS THE SMALLEST S/N RATIO.  ITWO IS THE LARGEST S/N.
C          ITHREE IS THE S/N LOOP INCREMENT.
C          A TYPICAL CARD 4 IS AS FOLLOWS...003009003...THIS WOULD
C          CAUSE FILTERS WITH S/N RATIOS OF 30,60,90 DECIBELS TO
C          BE DESIGNED.
C          FORMAT  3I3
        READ (5,260) IONE,ITWO,ITHREE
C
C     CARD5
C          NMR ARRAY TAKES ON THE VALUES OF NUM. ORDER FOR ARMA FILTERS.
C          EACH VALUE OF NMR SHOULD BE .LE. 128.
C          FORMAT 10I3
        READ (5,350) (NMR(I),I = 1,JJA)
C
C     CARD6
C          NEN ARRAY STORES VALUES OF DEN. ORDER FOR AR OR ARMA.
C          A MAX. OF 10 VALUES CAN BE STORED. EACH OF THESE MUST BE
C          .LE.256 FOR AR, AND .LE.128 FOR ARMA.
C          FORMAT 10I3
        READ (5,350) (NEN(I),I = 1,JJA)
C
        PI = 3.141592654
        T = 0.5
        PPP=0.5/T
C          T IS THE SAMPLING PERIOD.
C          PPP IS THE FOLDOVER FREQUENCY.
        MAUG = 1024
        NN = 512
C
C          THE USER SHOULD DEFINE VARIABLES APPLICABLE TO
C          THE TYPE OF FILTER TO BE DESIGNED.
C          THE VARIABLES FO,BWS,BWN,FQ... DEFINE THE IDEAL
C          POWER SPECTRAL DENSITY OF A LOWPASS FILTER.
C          THE FOLLOWING SECTION BRACKETED BY STARS NEEDS TO BE
C          CHANGED FOR OTHER TYPES OF FILTERS.
C          THIS SECTION COMPUTES THE COVARIANCE OF THE IDEAL P.S.D.
C          ***********************************************************
C
        BWQ=PPP-(BWN+BWS)
        FO=PPP-BWN/2.0
        FQ=(PPP-BWN+BWS)/2.0
C          COMPUTE COVARIANCE-RS=SIGNAL COV., PN=NOISE COV.
C          QN=TRANSITION BAND COVARIANCE
        PN(1) = 2.0 * BWN
        RS(1) = 2.0 * BWS
        QN(1) = 0.0
        DO 100 J = 2,257
        RS(J)=SIN(2.*PI*FLOAT(J-1)*T*BWS)/(PI*T*FLOAT(J-1))
        PN(J)=2.0*SIN(PI*BWN*FLOAT(J-1)*T)/(PI*FLOAT(J-1)*T)*COS(2.0*
       CPI*FO*FLOAT(J-1)*T)
        QN(J) = 0.0
  100   CONTINUE
C
C          ***********************************************************
C          AN=NOISE LEVEL.          AS=SIGNAL LEVEL(A VARIABLE)
C          AQ=TRANS. BAND LEVEL
C
        AN = 1.0
        DO 230 KK = IONE,ITWO,ITHREE
        AS = 10 *   * KK
        DO 230 NNA = 1,JJA
        NDEN = NEN(NNA)
        MAR = NDEN + 1
        IF (TOOL(1).EQ.1) GO TO 110
C          FOR ARMA DESIGNS, AR ORDER=256
        MAR = 257
  110   MR = MAR - 1
        R(1) = AS * RS(1) + AN * PN(1) + AQ * QN(1)
        DO 120 J = 2,MAR
        R(J) = AS * RS(J) + AN * PN(J) + AQ * QN(J)
```

```
C
C         R=TOTAL COVARIANCE VECTOR.  EQUALS SIGNAL+NOISE COV., WEIGHTED
C         BY AS AND AN.
C         ARRAY QN ALLOWS THE DESIGNER TO SPECIFY A TRANSITION
C         BAND COVARIANCE WITH WEIGHT AQ AND CENTER FREQ. FQ.
C
  120 CONTINUE
      BS = MAR - 1
      DO 130 LQ = 1,BS
      LQ1 = LQ + 1
      PP(LQ) =  - R(LQ1)
  130 CONTINUE
      CALL TPSLV (BS,R,PP,WAR,E)
C         SUBROUTINE TPSLV SOLVES TOEPLITZ SET OF EQUATIONS-
C         R*WAR=PP.
      JS = MAR
  140 JT = JS - 1
      WAR(JS)=WAR(JT)
      IF (JS.EQ.2) GO TO 150
      JS=JS-1
      GO TO 140
  150 CONTINUE
      DO 160 I = 1,257
      A(I) = 0.0
      Q(I) = 0.0
  160 CONTINUE
      WAR(1) = 1.0
C         GSQR=AR FILTER GAIN COEF. SQUARED
      GSQR = R(1)
      DO 170 JP = 2,MAR
      GSQR = GSQR + WAR(JP) * R(JP)
  170 CONTINUE
      Q(1) = GSQR *   * .5
C         GENERATE IMPULSE RESP. SEQUENCE FOR MULLIS/ROBERTS ALGORITHM.
      CALL IMPT (257,H,WAR,Q)
      IF (TOOL(1).NE.1) GO TO 180
C
C         THE FOLLOWING STARRED SECTION IS USED FOR AR DESIGNS ONLY.
C         ALTHOUGH AR(N)=ARMA(N,0), A TIME SAVINGS RESULTS
C         BY HAVING A SEPARATE AR DESIGN PROCEDURE.  THE AR
C         SOFTWARE IS USED FOR ARMA DESIGNS ALSO.
C         ***********************************************************
C
      CALL ARMAMG (Q,WAR,MAR,WSQ,PHASE,1)
      CALL SPEPLT (NN,WSQ,PHASE,HAR,PHAR,FREQ)
      WRITE (6,330)
      WRITE (6,310) NDEN,KK
      WRITE (6,330)
      WRITE (6,340) BWS,BWN
      WRITE (6,330)
      IF(TOOL(7).EQ.0) GO TO 175
      KKL = TOOL(2)
      CALL PLOT (HAR,PHAR,NN,KKL,TOOL,FREQ,PPP)
  175 CONTINUE
      WRITE (6,330)
      IF (TOOL(3).EQ.0) GO TO 180
      WRITE (6,290)
      WRITE (6,330)
      CALL PRINT(WAR,R,H,A,MAR,3)
C
C         ***********************************************************
  180 CONTINUE
C         GO TO 240 IF AN AR FILTER IS BEING DESIGNED.
      IF (TOOL(4).EQ.0) GO TO 230
      Q(1) = 0.0
      DO 190 MY = 1,257
      HQ(MY) = H(MY)
  190 CONTINUE
      IDIF = NEN(NNA) - NMR(NNA)
      JDIF = 0
      IF (IDIF.LT.0) JDIF = 1
      IF (IDIF.LT.0) IDIF =  - IDIF
      IF (IDIF.EQ.0) GO TO 200
```

```
C         GRAMMIAN SHIFT IMP. SEQ. ACCORDING TO DIFFERENCE IN NUM.
C         AND DEN. ORDER IN ARMA FILTER (SEE MULLIS/ROBERTS).
      CALL SHIFF (H,IDIF,MAR,JDIF)
  200 NFEN = NEN(NNA)
      MAR = NFEN + 1
      LSTAR = MAR - IDIF
      IF (JDIF.EQ.1) LSTAR = MAR + IDIF
C         IF NUM. ORDER .GT. DEN. ORDER, MODIFY COV. SEQ.
C         (SEE MULLIS/ROBERTS)
C         HQ IS UNSHIFTED IMP. SEQ.
      IF (JDIF.EQ.1) CALL COV (HQ,R,MAR,IDIF)
C         GENERATE ARMA COEFFICIENTS GIVEN IMP. AND COV. SEQUENCES.
      CALL MULRB (MAR,H,R,A)
      CALL NUMCF (Q,LSTAR,MAR,HQ,A)
      CALL IMPT (257,H,A,Q)
      WRITE (6,330)
      CALL ARMAMG (Q,A,MAR,WSQ,PHASE,LSTAR)
      CALL SPEPLT (NN,WSQ,PHASE,HAR,PHAR,FREQ)
      WRITE (6,330)
      WRITE (6,330)
      IF (IDIF.NE.0) GO TO 210
      NUMORD = NFEN
      GO TO 220
  210 CONTINUE
      NUMORD = LSTAR - 1
  220 CONTINUE
      WRITE (6,320) NFEN,NUMORD,KK
      WRITE (6,330)
      WRITE (6,340) BWS,BWN
      WRITE (6,330)
      KKK = TOOL(5)
      IF (TOOL(7).EQ.0) GO TO 225
      CALL PLOT (HAR,PHAR,NN,KKK,TOOL,FREQ,PPP)
  225 CONTINUE
      WRITE (6,330)
      IF (TOOL(6).EQ.0) GO TO 230
      WRITE (6,300)
      WRITE (6,330)
      IF (MAR.LT.LSTAR) MAR=LSTAR
      CALL PRINT (Q,A,R,H,MAR,4)
  230 CONTINUE
  240 CONTINUE
  250 FORMAT (1I1)
  260 FORMAT (3I3)
  270 FORMAT (10I1)
  280 FORMAT (3F3.2)
  290 FORMAT (9X,13H FILTER COEF.,5X,10H COV. SEQ.,12X,10H IMP. SEQ.)
  300 FORMAT (9X,11H NUM. COEF.,11X,11H DEN. COEF.,9X,10H COV. SEQ.,11X,
     110H IMP. SEQ.)
  310 FORMAT (13H AR FILTER N=,I3,5X,9H S/N=10**,I3)
  320 FORMAT (6H ARMA(,I3,1H,,I3,1H),5X,9H S/N=10**,I3)
  330 FORMAT (1H0)
  340 FORMAT (5H BWS=,F5.2,5H BWN=,F5.2)
  350 FORMAT (10I3)
      END
      SUBROUTINE MULRB(N,H,R,A)
C
C         COMPUTES DEN. COEFS. FOR ARMA.
C         A ARRAY CONTAINS THESE COEF.
C
      DIMENSION A(N),B(257),C(257)
      DIMENSION H(N),R(N)
      ALPH = R(1) - H(1) *  * 2
      A(1) = 1.0
      B(1) = 1.0/ALPH
      C(1) = H(1)/ALPH
      DELT = 1.0 + H(1) *  * 2/ALPH
      NSTAR = N - 1
      DO 100 I = 2,257
      B(I) = 0.0
      C(I) = 0.0
  100 CONTINUE
```

```
      DO 170 I = 1,NSTAR
      GMA = 0.0
      BET = 0.0
      DO 110 K = 1,I
      BET = BET + A(K) * R(I + 2 - K)
      GMA = GMA - A(K) * H(I + 2 - K)
  110 CONTINUE
      THET = BET * B(1) + GMA * C(1)
      PHI = BET * C(1) + GMA * DELT
      ALPH = ALPH - BET * THET - GMA * PHI
      IF (ALPH) 130,120,130
  120 WRITE (6,190)
      GO TO 180
  130 DELT = DELT + (PHI *   * 2)/ALPH
      IPLUS = I + 1
      DO 140 K = 1,IPLUS
      A(K) = A(K) - BET * B(I + 2 - K) - GMA * C(I + 2 - K)
  140 CONTINUE
      DO 150 K = 1,IPLUS
      B(K) = B(K) - (THET/ALPH) * A(I + 2 - K)
  150 CONTINUE
      DO 160 K = 1,IPLUS
      C(K) = C(K) - (PHI/ALPH) * A(I + 2 - K)
  160 CONTINUE
      A(I + 2) = 0.0
      B(I + 2) = 0.0
      C(I + 2) = 0.0
  170 CONTINUE
  180 RETURN
  190 FORMAT(5X,15H ERROR IN MULR3)
      END
      SUBROUTINE ARMAMG(Q,A,N,WSQR,PH,LSTAR)
C
C        COMPUTES ARMA MAG. AND PHASE GIVEN NUM. AND DEN. COEF.
C        THE Q ARRAY CONTAINS NUM. COEFS.
C        THE A ARRAY CONTAINS DEN. COEFS.
C        LSTAR=NUM. ORDER +1
C        N=DEN. ORDER +1
C
      DIMENSION PH(512),WSQR(512),WSQ1(512),WSQ2(512),Q(N),A(N)
      COMPLEX DATA1(1024),DATA2(1024)
      SUM1 = 0.
      SUM2 = 0.
      DO 100 I = 1,N
      SUM2 = SUM2 + A(I)
  100 CONTINUE
      DO 110 I = 1,LSTAR
      SUM1 = SUM1 + Q(I)
  110 CONTINUE
      DO 120 I = 1,N
      A(I) = A(I)/SUM2
  120 CONTINUE
      DO 130 I = 1,LSTAR
      Q(I) = Q(I)/SUM1
  130 CONTINUE
      DO 140 I = 1,N
      DATA1(I)=CMPLX(A(I),0.0)
  140 CONTINUE
      DO 150 I = 1,LSTAR
      DATA2(I)=CMPLX(Q(I),0.0)
  150 CONTINUE
      MP = N + 1
      DO 160 I = MP,1024
      DATA1(I)=CMPLX(0.,0.)
  160 CONTINUE
      MP = LSTAR + 1
      DO 170 I = MP,1024
      DATA2(I)=CMPLX(0.,0.)
  170 CONTINUE
```

```
      CALL FOURT (DATA1,1024,1, - 1,0,0.0)
      CALL FOURT (DATA2,1024,1, - 1,0,0.0)
      DO 180 I = 1,512
      PH(I)=57.2957795*((ATAN2(AIMAG(DATA2(I)),REAL(DATA2(I))))-
     CATAN2(AIMAG(DATA1(I)),REAL(DATA1(I))))
      PH(I) =  - PH(I)
      WSQ1(I) = REAL(DATA1(I) * CONJG(DATA1(I)))
      WSQ2(I) = REAL(DATA2(I) * CONJG(DATA2(I)))
  180 CONTINUE
      DO 200 I = 1,512
      IF (WSQ1(I)) 190,200,190
  190 WSQR(I) = WSQ2(I)/WSQ1(I)
  200 CONTINUE
      RETURN
      END
      SUBROUTINE SPEPLT(NN,WSQ,PHASE,HAR,PHAR,FREQ)
C
C        COMPUTES A SEQUENCE OF MAG. AND PHASE POINTS TO BE SENT TO PLOT.
C        MAG. SQUARED VECTOR IS WSQ.
C        HAR VECTOR CONTAINS WSQ IN DECIBELS.
C        PHAR CONTAINS A SEQUENCE OF PHASE POINTS.
C
      DIMENSION WSQ(NN),PHASE(NN),HAR(NN),PHAR(NN),FREQ(NN)
      DO 100 I = 1,NN
      HAR(I) = 10. * ALOG10(WSQ(I))
      PHAR(I) =  - PHASE(I)
      FREQ(I) = FLOAT(I - 1)/512
  100 CONTINUE
      RETURN
      END
      SUBROUTINE PRINT(A,B,C,D,NA,NB)
C
C        PRINTS OUT ANY ONE,TWO,THREE,OR FOUR 1-DIMENSIONAL,EQUAL
C        LENGTH ARRAYS.  NA IS THE NUMBER OF ARRAY ELEMENTS TO BE PRINTED.
C
      DIMENSION OUT(4,257)
      DIMENSION A(257),B(257),C(257),D(257)
      DO 100 J = 1,4
      DO 100 I = 1,257
      OUT(J,I) = 0.0
  100 CONTINUE
      DO 140 I = 1,NA
      IF (NB.EQ.1) GO TO 130
      IF (NB.EQ.2) GO TO 120
      IF (NB.EQ.3) GO TO 110
      OUT(4,I) = D(I)
  110 OUT(3,I) = C(I)
  120 OUT(2,I) = B(I)
  130 OUT(1,I) = A(I)
  140 CONTINUE
      DO 150 I = 1,NA
      WRITE (6,160) I,(OUT(J,I),J = 1,4)
  150 CONTINUE
      RETURN
  160 FORMAT (2X,I3,5X,1P4E20.7)
      END
      SUBROUTINE TPSLV(N,R,PS,W,E)
C
C        ROUTINE WRITTEN BY DAVE FARDEN, COLORADO STATE UNIV.
C
      DIMENSION R(N),PS(N),W(N),E(N)
      NM1 = N - 1
      CINV = 1.0/R(1)
      DO 100 I = 1,NM1
      R(I) = CINV * R(I + 1)
  100 PS(I) = CINV * PS(I)
      PS(N) = CINV * PS(N)
      W(1) = PS(1)
      E(1) =  - R(1)
      AMB = 1.0 - R(1) * R(1)
```

```
      DO 160 I = 1,NM1
      I1 = I + 1
      THET = PS(I1)
      ET =  - R(I1)
      DO 110 L = 1,I
      LI = I - L + 1
      THET = THET - W(L) * R(LI)
  110 ET = ET - R(L) * E(LI)
      C1 = THET/AMB
      C2 = ET/AMB
      I1 = I + 1
      IBAR = I1/2
      DO 120 L = 1,I
      LI = I1 - L
  120 W(L) = W(L) + C1 * E(LI)
      DO 130 L = 1,IBAR
      LI = I1 - L
      C3 = E(L)
      E(L) = E(L) + C2 * E(LI)
  130 E(LI) = E(LI) + C2 * C3
      IF (IBAR - I/2) 150,150,140
  140 E(IBAR) = C3 * (1.0 + C2)
  150 W(I1) = C1
      E(I1) = C2
  160 AMB = AMB - C2 * ET
C     UNNORMALIZE R AND PS
      J = N
      C = 1.0/CINV
      DO 170 I = 1,NM1
      R(J) = C * R(J - 1)
      PS(I) = C * PS(I)
  170 J = J - 1
      PS(N) = C * PS(N)
      R(1) = C
      RETURN
      END
      SUBROUTINE PLOT(HAR,PHAR,NN,N,TOOL,FREQ,PPP)
C
C        MAPA AND MAPM ARE PLOTTING ROUTINES AVAILABLE AT COL. STATE U.
C        MAPA IS A PAPER PLOT.
C        MAPM IS A MICROFILM PLOT.
C
      INTEGER TOOL(10)
      DIMENSION HAR(NN),PHAR(NN)
      DIMENSION FREQ(NN)
      INTEGER XT(8),YT(8),PT(8),MT(7)
      XT(1) = 10HFREQ(HZ)
      XT(2) = 10H
      YT(1) = 10HMAGNSQ(DB)
      YT(2) = 10H
      PT(1) = 10HPHASE(DGR)
      PT(2) = 10H
      DO 100 I = 3,8
      XT(I) = 10H
      YT(I) = 10H
      PT(I) = 10H
  100 CONTINUE
      MT(1) = 10HFILTER
      MT(2) = 10HDESIGN
      MT(3) = 10H
      MT(4) = 10H
      MT(5) = 10H
      MT(6) = 10H
      MT(7) = 4H
      CALL MAPA (1,FREQ,HAR,1,NN,0.0,PPP, - 120.0,30.0,XT,YT,MT, - 1)
      CALL MAPA (2,FREQ,HAR,1,NN,0.0,PPP, - 120.0,30.0,XT,YT,MT, - 1)
      CALL MAPA (4,FREQ,HAR,1,NN,0.0,PPP, - 120.0,30.0,XT,YT,MT, - 1)
      IF (TOOL(8).EQ.0) GO TO 110
      CALL MAPM (1,FREQ,HAR,1,NN,0.0,PPP, - 120.0,30.0,XT,YT,MT, - 1)
      CALL MAPM (2,FREQ,HAR,1,NN,0.0,PPP, - 120.0,30.0,XT,YT,MT, - 1)
      CALL MAPM (4,FREQ,HAR,1,NN,0.0,PPP, - 120.0,30.0,XT,YT,MT, - 1)
```

```
  110 IF (N.EQ.0) GO TO 120
      CALL MAPA(1,FREQ,PHAR,1,NN,0.0,PPP,-360.,360.,XT,YT,MT,-1)
      CALL MAPA(2,FREQ,PHAR,1,NN,0.0,PPP,-360.,360.,XT,YT,MT,-1)
      CALL MAPA(4,FREQ,PHAR,1,NN,0.0,PPP,-360.,360.,XT,YT,MT,-1)
      IF (TOOL(8).EQ.0) GO TO 120
      CALL MAPM(1,FREQ,PHAR,1,NN,0.0,PPP,-360.,360.,XT,YT,MT,-1)
      CALL MAPM(2,FREQ,PHAR,1,NN,0.0,PPP,-360.,360.,XT,YT,MT,-1)
      CALL MAPM(4,FREQ,PHAR,1,NN,0.0,PPP,-360.,360.,XT,YT,MT,-1)
  120 RETURN
      END
      SUBROUTINE IMPT(N,H,A,Q)
C
C        CONPUTES ARMA IMP. RESPONSE GIVEN COEFFICIENTS.
C        N VALUES OF THE RESPONSE ARE COMPUTED AND STORED IN H.
C        N.LE.257
C
      DIMENSION H(N),A(N),Q(N)
      DO 130 I = 1,N
      H(I) = 0.0
      SUM = 0.0
      DO 110 J = 2,N
      IF (I - J + 1) 120,120,100
  100 SUM = SUM - A(J) * H(I - J + 1)
  110 CONTINUE
  120 CONTINUE
      H(I) = Q(I) + SUM
  130 CONTINUE
      RETURN
      END
      SUBROUTINE SHIFF(H,K,NL,NX)
C
C        THIS ROUTINE PERFORMS A GRAMMIAN SHIFT ON THE IMPULSE
C        SEQUENCE H.  K IS THE NUMBER OF SHIFTS TO BE MADE.
C        NL IS THE LENGTH OF THE H SEQUENCE AND IT MUST BE CONSISTENT
C        WITH THE LENGTH OF THE H SEQ. GENERATED BY IMPT.
C        NX =1 FOR LEFT SHIFTS AND 0 FOR RIGHT SHIFTS.
C
      DIMENSION H(NL)
      IF (NX.EQ.1) GO TO 130
      I = NL
  100 H(I) = H(I - K)
      I = I - 1
      IF (I.GT.K) GO TO 100
      I = 1
      IF (K) 120,120,110
  110 IF (I.GT.K) GO TO 120
      H(I) = 0.0
      I = I + 1
      GO TO 110
  120 CONTINUE
      GO TO 150
  130 JX = 257 - K
      DO 140 I = 1,JX
      H(I) = H(I + K)
  140 CONTINUE
  150 CONTINUE
      RETURN
      END
      SUBROUTINE NUMCF(Q,LSTAR,MAR,HQ,A)
C
C        GENERATES ARMA NUM. COEFS. GIVEN DEN. COEFS. AND IMP.
C        SEQ.(UNSHIFTED).
C
      DIMENSION Q(LSTAR),HQ(MAR),A(MAR)
      NSTAR = MAR
      DO 120 I = 1,LSTAR
      DO 110 J = 1,NSTAR
      IF (I - J + 1) 120,120,100
  100 Q(I) = Q(I) + A(J) * HQ(I - J + 1)
  110 CONTINUE
  120 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE COV(HQ,R,MAR,IDIF)
C
C        MODIFIES COV. SEQ. FOR A LEFT SHIFTED IMP. SEQ.
C
      DIMENSION R(257),HQ(257)
      DO 110 I = 1,MAR
      SUM = 0.0
      DO 100 J = 1,IDIF
      SUM = SUM - HQ(J) * HQ(I + J - 1)
  100 CONTINUE
      R(I) = R(I) + SUM
  110 CONTINUE
      RETURN
      END
```

References

[1] J. Makhoul, "Linear Prediction: A Tutorial Review," Proc. IEEE, 63, pp. 561-580 (April 1975).

[2] C. T. Mullis and R. A. Roberts, "The Use of Second-Order Information in the Approximation of Discrete-Time Linear Systems," IEEE Trans. ASSP, ASSP-24, pp. 226-238 (June 1976).

[3] C. S. Burrus and T. W. Parks, "Time Domain Design of Recursive Digital Filters," IEEE Trans. Audio Electroacoust., AU-18, pp. 137-141 (June 1970).

[4] F. N. Cornett, "First and Second Order Techniques for the Design of Recursive Filters," Ph.D. dissertation, Colorado State University, Ft. Collins, Colo. (1976).

[5] D. C. Farden and L. L. Scharf, "Statistical Design of Nonrecursive Digital Filters," IEEE Trans. ASSS, ASSP-22, pp. 188-196 (June 1974).

[6] E. Parzen, "Multiple Time Series Modelling," in Multivariate Analysis II, P. Krishnaiah (Ed.), Academic Press: New York (389-409).

[7] W. C. Kellogg, "Time Domain Design of Nonrecursive Least Mean-Square Digital Filters," IEEE Trans. Audio Electroacoust., AU-20, pp. 155-158 (June 1972).

[8] D. C. Farden and L. L. Scharf, "Authors Reply to 'Comments on Statistical Design of Nonrecursive Digital Filters'," IEEE Trans. ASSP, ASSP-23, pp. 495-496 (Oct. 1975).

[9] U. Grenander and G. Szegö, "Toeplitz Forms and Their Applications," Univ. of California Press: Berkeley (1958).

TABLE OF FIGURES

Figure 1. An Ideal Lowpass Spectrum: (a) The Spectrum, (b) The Spectrum Scaled and Periodically-Extended.

Figure 2. AR(N_1) Approximations to Ideal Lowpass Spectra: N_1 = 32, 64, 256.

Figure 3. Flowchart for Design Procedure.

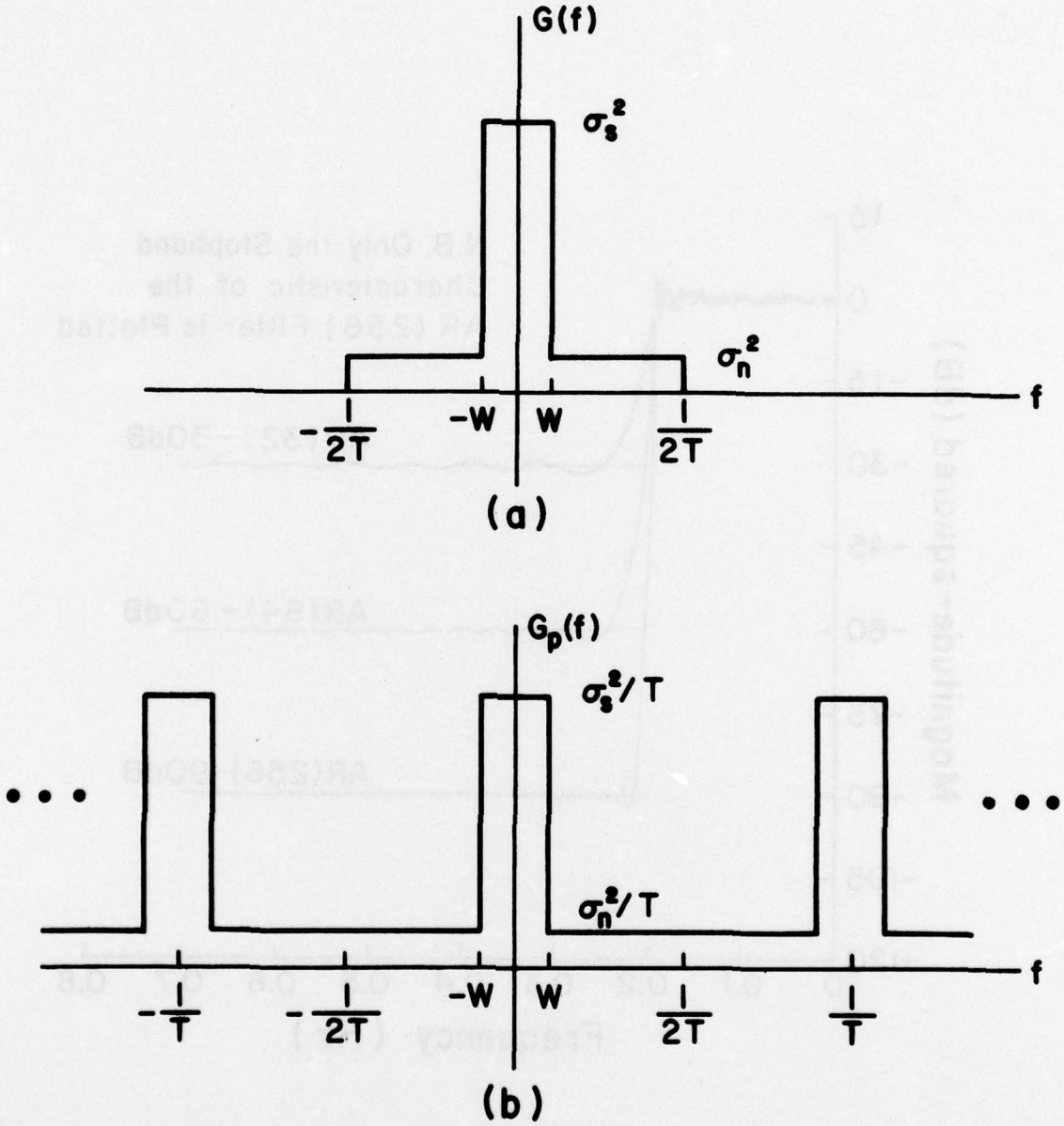
Figure 4. ARMA(N,N) Approximations to AR(N_1): (a) N_1 = 64, N = 8, 30dB Spectrum; (b) N_1 = 256, N = 32, 90dB Spectrum.

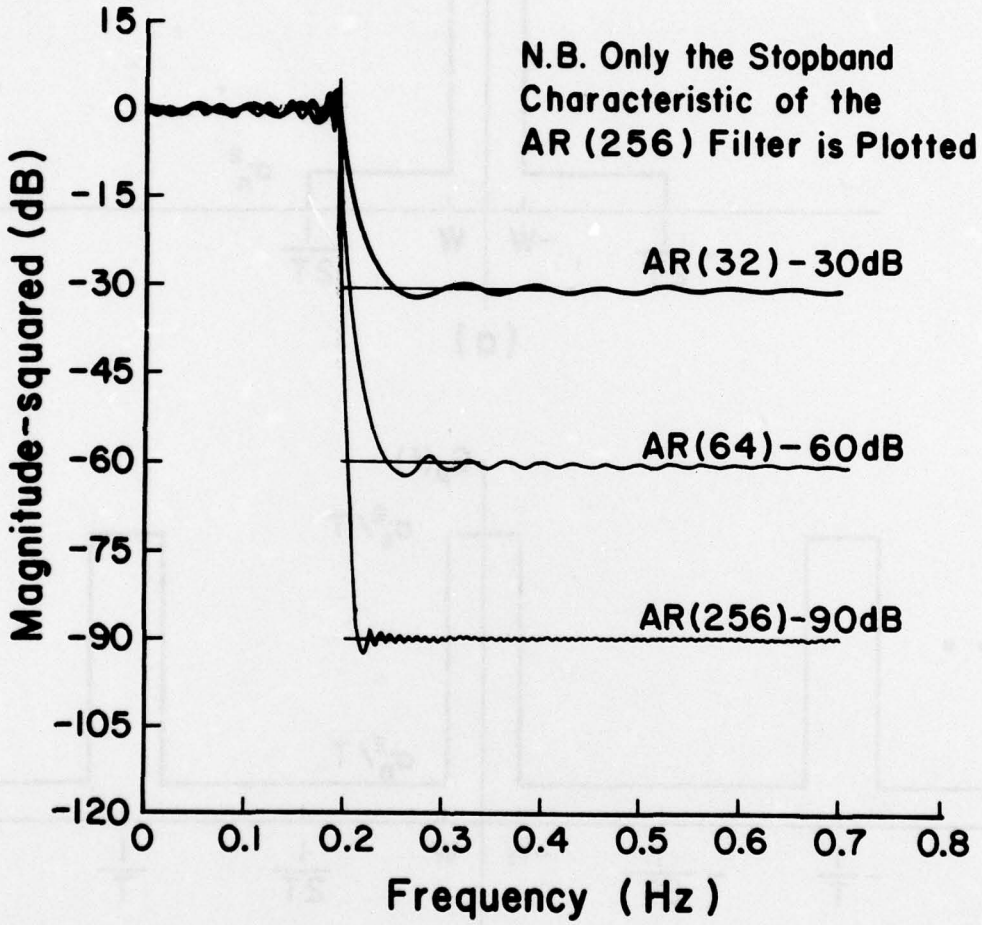
Figure 5. ARMA(8,M) Approximations to Ideal Lowpass Spectra: (a) M = 4; 30, 60, and 90dB Spectra; (b) M=12; 30, 60, and 90dB Spectra.

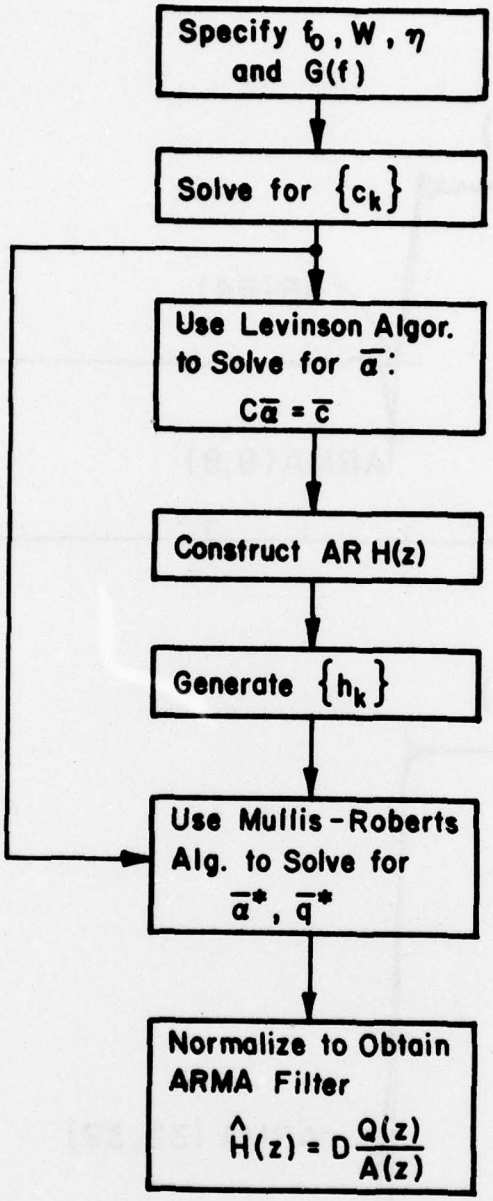
Figure 6. ARMA(16,M) Approximations to a 60dB Spectrum: M = 8, 16, 24.

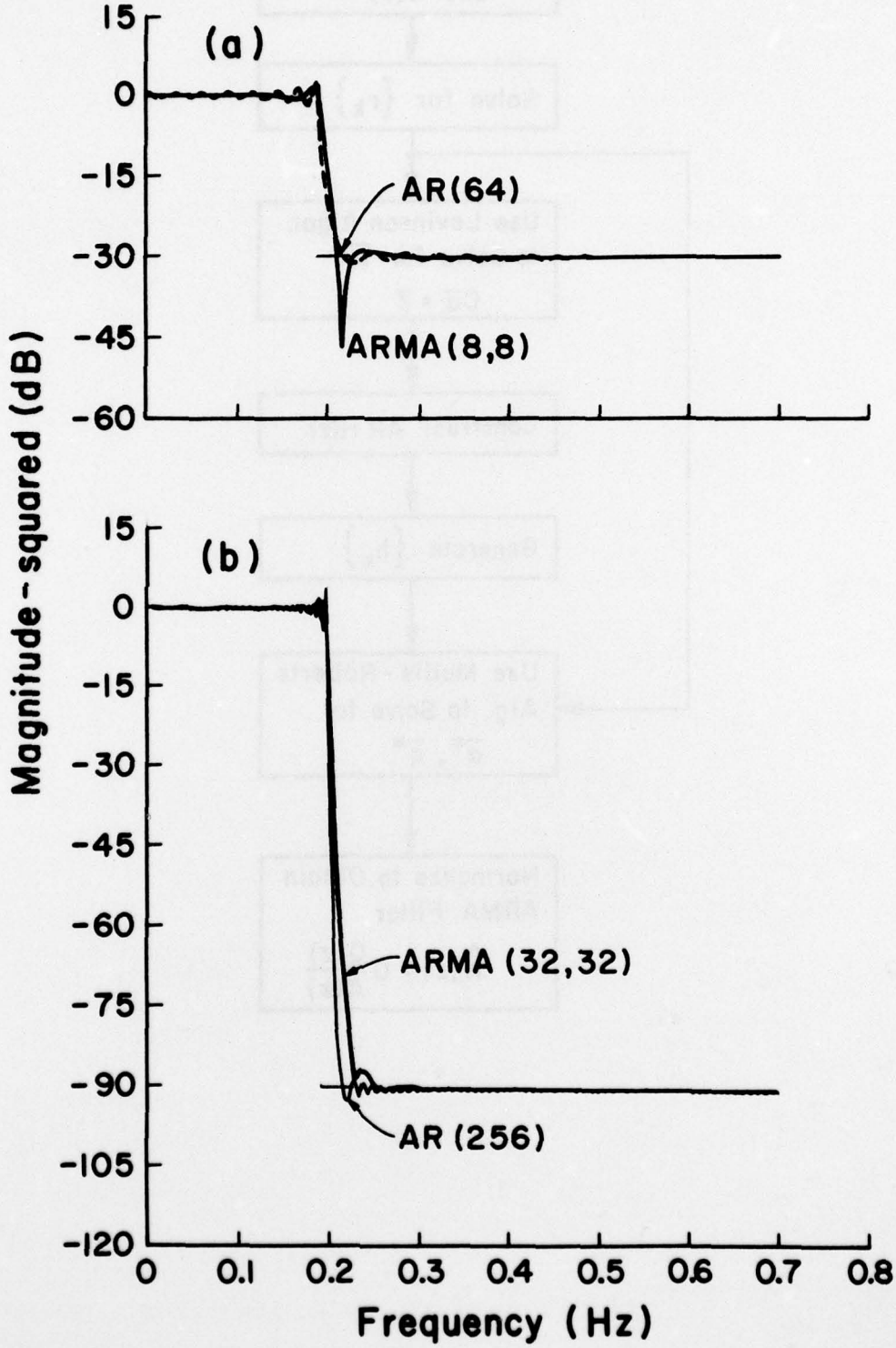
Figure 7. ARMA(8,M) Approximations to Ideal Lowpass Spectra: Comparisons with Bilinear Butterworth and Chebyshev Designs.

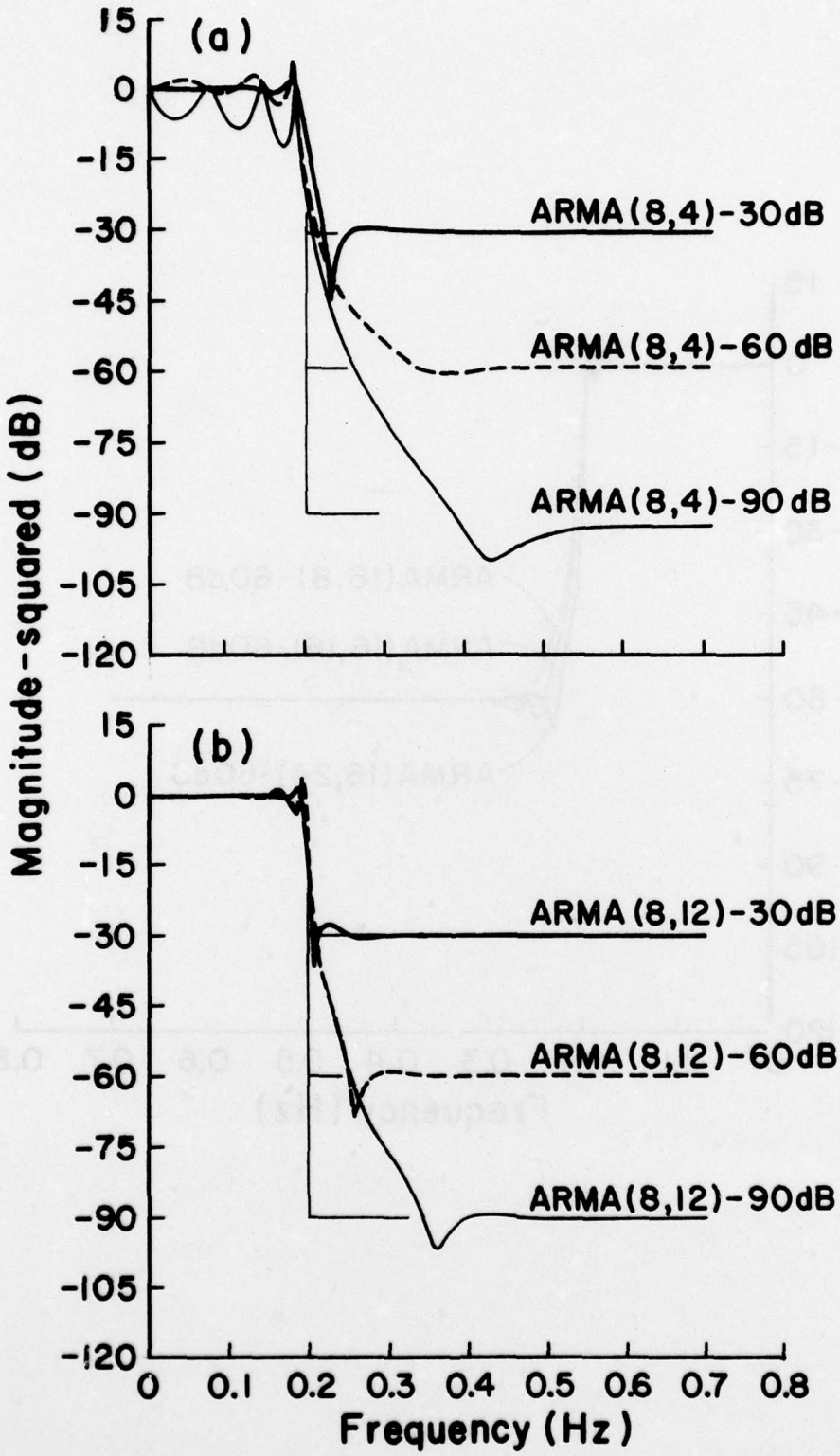
Figure 8. ARMA(16,M) Approximations to Ideal Lowpass Spectra: Comparisons with Bilinear Butterworth and Chebyshev Designs.

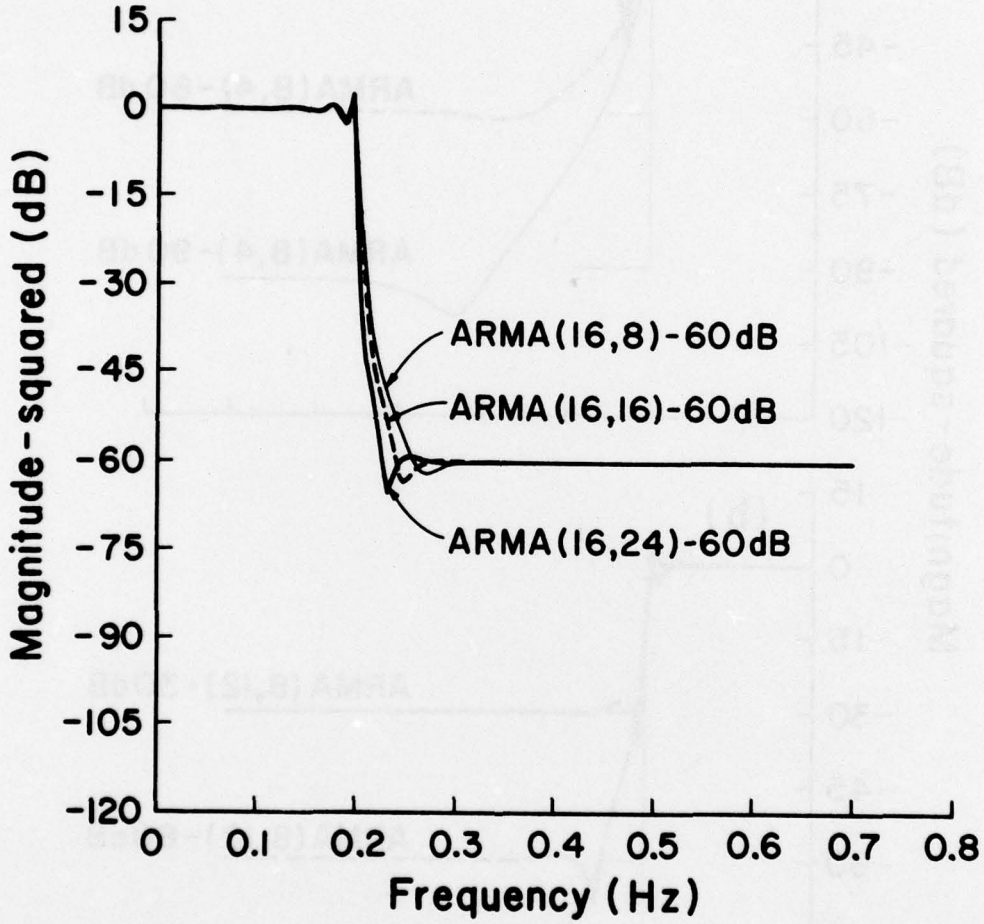

G(f)
σ_s^2
σ_n^2
f
$-\frac{1}{2T}$
-W
W
$\frac{1}{2T}$
(a)
$G_p(f)$
σ_s^2/T
σ_n^2/T
f
$-\frac{1}{T}$
$-\frac{1}{2T}$
-W
W
$\frac{1}{2T}$
$\frac{1}{T}$
(b)

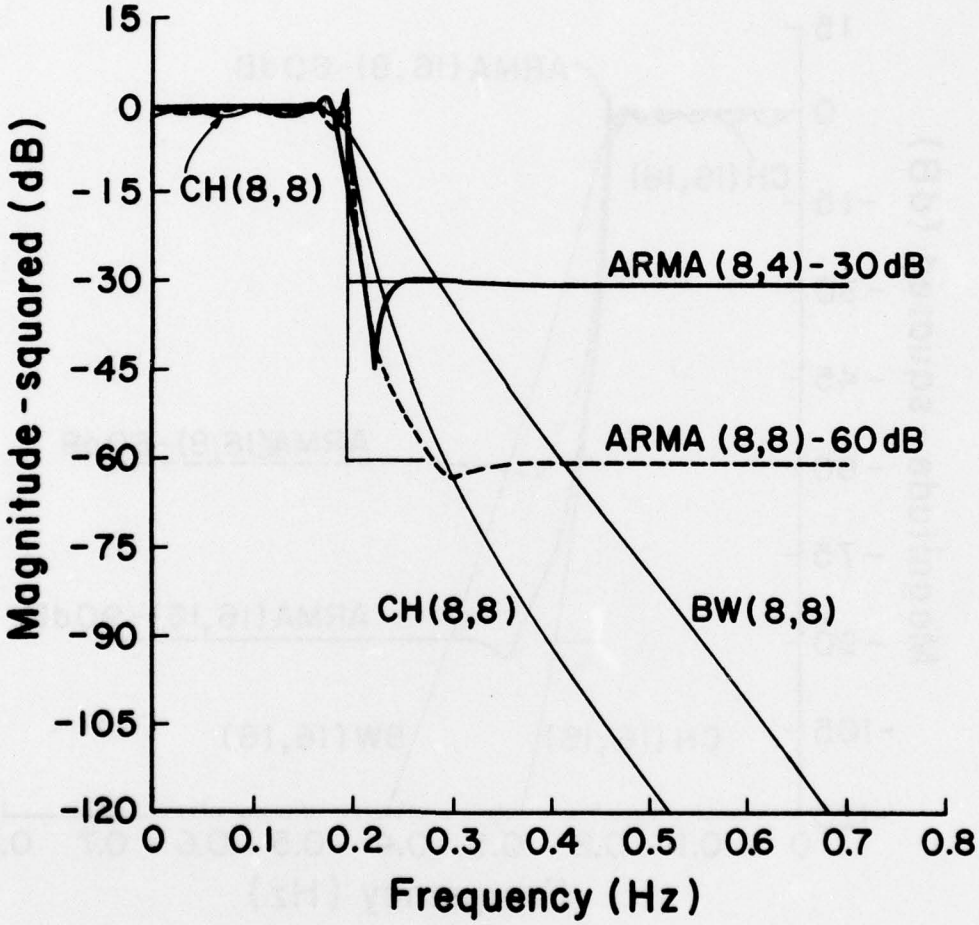

N.B. Only the Stopband Characteristic of the AR (256) Filter is Plotted
AR(32) - 30dB
AR(64) - 60dB
AR(256) - 90dB
Magnitude-squared (dB)
15
0
-15
-30
-45
-60
-75
-90
-105
-120
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Frequency (Hz)

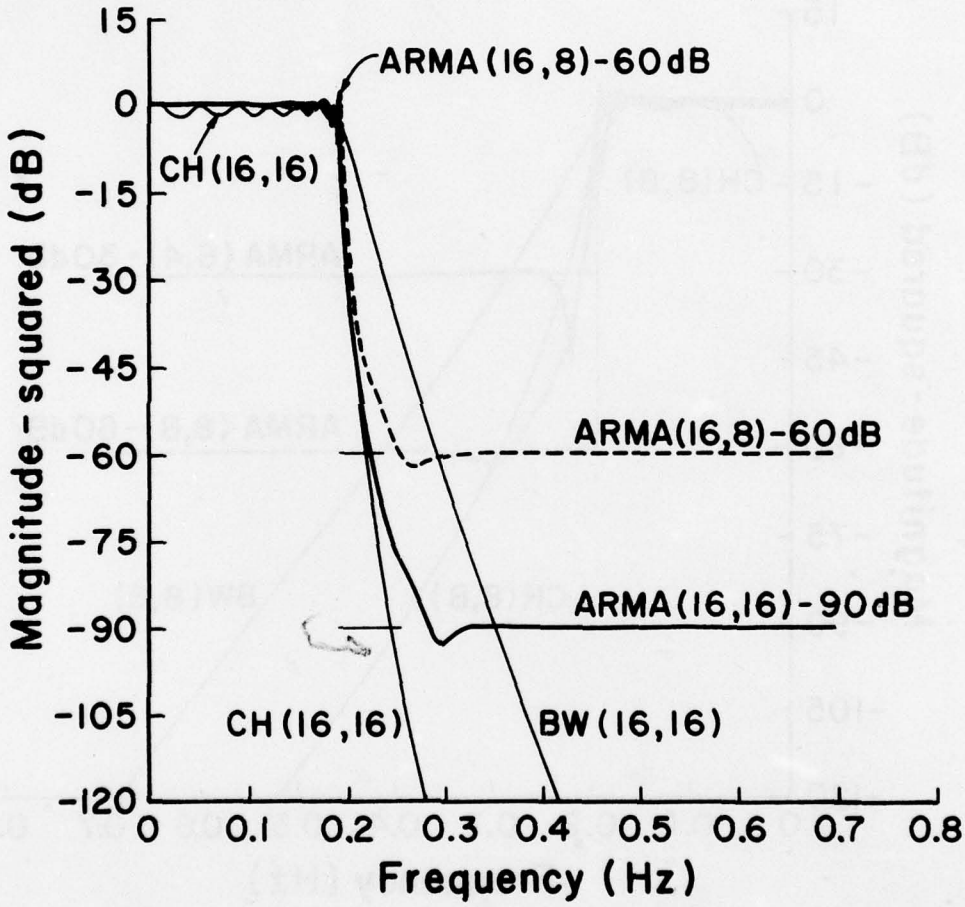

Specify f_0, W, η and G(f)
Solve for $\{c_k\}$
Use Levinson Algor. to Solve for $\bar{\alpha}$: $C\bar{\alpha} = \bar{c}$
Construct AR H(z)
Generate $\{h_k\}$
Use Mullis-Roberts Alg. to Solve for $\bar{\alpha}^*$, $\bar{q}^*$
Normalize to Obtain ARMA Filter $\hat{H}(z) = D\frac{Q(z)}{A(z)}$


(a)
(b)
AR(64)
ARMA(8,8)
ARMA (32,32)
AR (256)
Magnitude-squared (dB)
Frequency (Hz)
15
0
-15
-30
-45
-60
-75
-90
-105
-120
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8


(a)
(b)
ARMA(8,4)-30dB
ARMA(8,4)-60 dB
ARMA(8,4)-90 dB
ARMA(8,12)-30dB
ARMA(8,12)-60dB
ARMA(8,12)-90dB
Magnitude-squared (dB)
15
0
-15
-30
-45
-60
-75
-90
-105
-120
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Frequency (Hz)


15
0
-15
-30
-45
-60
-75
-90
-105
-120
Magnitude-squared (dB)
ARMA(16,8)-60dB
ARMA(16,16)-60dB
ARMA(16,24)-60dB
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Frequency (Hz)


15
0
-15
-30
-45
-60
-75
-90
-105
-120
Magnitude-squared (dB)
CH(8,8)
ARMA (8,4)-30dB
ARMA (8,8)-60dB
CH(8,8)
BW(8,8)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Frequency (Hz)


ARMA (16,8)-60dB
CH(16,16)
ARMA(16,8)-60dB
ARMA(16,16)-90dB
CH(16,16)
BW(16,16)
Magnitude-squared (dB)
15
0
-15
-30
-45
-60
-75
-90
-105
-120
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Frequency (Hz)

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: ONR Technical Report #21

2. GOVT ACCESSION NO.: (14)

3. RECIPIENT'S CATALOG NUMBER: TR-21 (ONR)

4. TITLE (and Subtitle): (6) Statistical Design of Autoregressive-Moving Average Digital Filters

5. TYPE OF REPORT & PERIOD COVERED: (9) Technical Report

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s): (10) Louis L. Scharf, James C. Luby

8. CONTRACT OR GRANT NUMBER(s): (15) N00014-75-C-0518

9. PERFORMING ORGANIZATION NAME AND ADDRESS:
Department of Electrical Engineering
Colorado State University
Fort Collins, CO 80523

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS:
Office of Naval Research
Statistics and Probability Branch
Arlington, VA 22217

12. REPORT DATE: (11) April 1977

13. NUMBER OF PAGES: 40 (12) 43p.

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):
Office of Naval Research
Statistics and Probability Branch
Arlington, VA 22217

15. SECURITY CLASS. (of this report): Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

406 434

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Autoregressive-moving average filters; autoregressive filters; Levinson algorithm; unit pulse sequence; covariance sequence; filter approximation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Procedures are presented for the systematic design of digital filters that contain poles and zeros. The procedures are simple, fast, and effective. All of the important algorithms are of the Levinson-type. The first key idea in the paper is that one may begin a design by posing a linear prediction problem for a stochastic sequence. The second is that a high-order "whitening" filter may be constructed for this sequence and "inverted" to yield a high-order all-pole filter whose spectrum approximates NEXT Page

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Abstract (cont.)

the spectrum of the stochastic sequence. The third key idea is that the all-pole filter may be used to generate consistent unit pulse and covariance sequences for use in the Mullis-Roberts algorithm. This algorithm is then used to obtain a low-order digital filter with poles and zeros that approximates the high-order all-pole filter. The results demonstrate that the Mullis-Roberts algorithm, together with the design philosophy of this paper, may be used with profit to reduce filter complexity and to design spectrum-matching digital filters.