THE DESIGN OF EQUIRIPPLE MATRIX FILTERS

Craig S. MacInnes

Applied Research Laboratories
University of Texas, Austin, Texas, 78713-8029

ABSTRACT

This paper presents a procedure for deriving linear filters which are based on matrix-vector multiplication instead of linear convolution and which can be designed to match the frequency responses of linear equiripple FIR filters. The magnitude of the matrix filter response is matched to the magnitude response of a given linear FIR filter by solving a set of nonlinear equations numerically using Broyden's method.

1. INTRODUCTION

Let $g = g_0, g_1, \ldots, g_{N-1}$ be a sequence of complex numbers which represent a set of discrete data points. Let $f = f_0, f_1, \ldots, f_{M-1}$ represent the coefficients of a linear FIR discrete filter. The operation of filtering the data sequence, g can be implemented as the linear convolution $h = f * g$ of g and the filter f, where

$$h_n = \sum_{k=0}^{M-1} f_k g_{[n-k]}, \quad n = 0, \ldots, N + M - 2.$$

where $[n-k] = 0$ when $n - k < 0$ and when $n - k > N - 1$. Let the sequence, g be written as a column vector of length N and let a matrix $A \in \mathbb{C}^{N \times N}$ be defined. The vector h defined by the matrix-vector product $h = Ag$ then represents a sequence of length N. The operation is linear, i.e. for sequences $g_1,\ g_2 \in \mathbb{C}^N$ and scalars $\alpha_1,\ \alpha_2 \in \mathbb{C}$, $A(\alpha_1 g_1 + \alpha_2 g_2) = \alpha_1 A g_1 + \alpha_2 A g_2$. Thus the matrix-vector multiplication can be interpreted as a linear filtering of the data sequence, g with the property (not shared by linear convolution filters) that the lengths of the initial and final sequences are the same. Since the rows of A are not equal and the entire data vector, g is required before the output Ag can be obtained, the matrix filter will in general be both time varying and noncausal.

The terms *convolution filter* and *matrix filter* will be used in order to differentiate between a linear filter implemented using linear convolution and a linear filter implemented using matrix-vector multiplication, respectively. Sequences of length N will be identified with column vectors of length N. The vector $f = [f_0, f_1, \ldots, f_{N-1}]^T$ representing the coefficients of a discrete filter is first determined using a conventional technique and then a matrix, $A \in \mathbb{C}^{N \times N}$ is designed which implements the filtering operation using matrix-vector multiplication. The design criteria are (1) the approximation of the magnitude of the frequency response of the given convolution filter and (2) the invertibility of the matrix filter.

In order to find the frequency response of a filter of length N at frequency ω_n, the magnitude of the filter response to the input sequence $v(\omega_n)$ is found. Define a *Fourier N-vector*, $v(\omega) = \left[1\ e^{\mathrm{j}\omega}\ \cdots\ e^{\mathrm{j}\omega(N-1)}\right]^T$, where $\mathrm{j} = \sqrt{-1}$ and $\omega \in [-\pi, \pi]$. If the filter f and sequence g are both of length N then there will be only one element in the output sequence $h = f * g$ where g and f have *full overlap*, i.e. where each element f_i of f multiplies an element g_j of g. This element is $h_{N-1} = (f * g)_{N-1} = f_{N-1} g_0 + f_{N-2} g_1 + \cdots + f_0 g_{N-1}$. To evaluate the response of the filter f at frequency $\omega = \omega_n$, let $g = v(\omega_n)$. Then

$$\begin{aligned}(f * g)_{N-1} &= \sum_{k=0}^{N-1} f_k\ g_{[N-1-k]} \\ &= e^{\mathrm{j}\omega_n(N-1)} \left(f_0 + \cdots + f_{N-1} e^{-\mathrm{j}\omega_n(N-1)}\right).\end{aligned}$$

The squared magnitude response of the filter A will be defined to be

$$\|Av(\omega)\|_2^2 = v(\omega)^H A^H A v(\omega), \quad -\pi \le \omega \le \pi.$$

The edge effects which occur when two sequences are convolved served to motivate research into linear filtering methods for short data sequences which preserve the length of the input data sequence. In [1] the idea of using matrix-vector multiplication to perform the filtering operation was investigated. Let a nonnegative real-valued function $\mu(\omega)$ representing a magnitude response be defined on $[-\pi, \pi]$. The design procedure

Email: macinnescs@ieee.org

used a convex optimization algorithm to achieve:

$$\|Av(\omega) - \mu(\omega)v(\omega)\| \leq \tau(\omega).$$

at prespecified frequencies $\omega = \omega_1, \ldots, \omega_p$, where the error tolerance at frequency ω_p was $\tau(\omega_p)$ and where the performance using both the sup $-$norm and $2-$norm were assessed. It was found by experiment that the computed matrix filter, $A \in \mathbb{C}^{N\times N}$ occasionally had more attenuation in the stopband than the corresponding convolution filter. In particular, an example of a Hilbert transformer was given for which the mean-squared error of the matrix filter was less than the filter designed using MATLAB functions. It was also found by experiment that low rank matrix filters gave results which were similar to full rank matrix filters. In order to decrease the run time of the optimization algorithm the matrix filter, A was represented in factored form as $A = WW^H$, $W \in \mathbb{C}^{N\times r}$ with r small compared to N. Thus, only Nr matrix elements rather than N^2 elements had to be determined, and the complexity of the matrix-vector product was decreased as well, since $Av = W(W^H v)$ takes only $O(Nr)$ instead of $O(N^2)$ complex operations. It will be shown, however, that the use of rank deficient matrix filters can and should be avoided.

In this paper a different perspective than in [1] is taken. A first step toward an explicit method for designing improved performance matrix equiripple filters is to develop a method for producing matrix filters which reproduce the frequency responses of FIR equiripple filters. A method for doing this is given here. This paper is organized as follows. In sections 2 and 3 the theory for the design of a matrix filter is presented. Section 4 discusses the resulting set of nonlinear equations which must be solved. The implementation of an equiripple matrix filter is given in section 5, followed by conclusions in section 6 .

2. THE LINEAR PROJECTION OPERATOR, T_M.

Consider a FIR filter, $f = [f_0\ f_1\ \cdots\ f_{N-1}]^T$ and define an N-vector, $c = [c_0\ c_1\ \ldots\ c_{N-1}]^T$, where $c_k = \bar{f}_{N-k-1},\ \ k = 0, \ldots, N-1$. Let $v(\omega)$ be a Fourier vector. The squared magnitude response of the filter, f is then

$$|(f * v(\omega))_{N-1}|^2 = \left|c^H v(\omega)\right|^2 = v(\omega)^H c\ c^H v(\omega). \quad (1)$$

Since $v(\omega)$ is a Fourier vector, (1) can be written as the rational function

$$r(z) = \bar{a}_{N-1}z^{-N+1} + \cdots + \bar{a}_1 z^{-1} + a_0 + \cdots + a_{N-1}z^{N-1}. \quad (2)$$

For example, when $c = [c_0\ c_1\ c_2\ c_3]^T$ equation (2) can be written as

$$r(z) = v(\omega)^H c\ c^H v(\omega) =$$
$$[1\quad z^{-1}\quad z^{-2}\quad z^{-3}] \begin{bmatrix} |c_0|^2 & c_0\bar{c}_1 & c_0\bar{c}_2 & c_0\bar{c}_3 \\ c_1\bar{c}_0 & |c_1|^2 & c_1\bar{c}_2 & c_1\bar{c}_3 \\ c_2\bar{c}_0 & c_2\bar{c}_1 & |c_2|^2 & c_2\bar{c}_3 \\ c_3\bar{c}_0 & c_3\bar{c}_1 & c_3\bar{c}_2 & |c_3|^2 \end{bmatrix} \begin{bmatrix} 1 \\ z \\ z^2 \\ z^3 \end{bmatrix}$$
$$= \bar{a}_3 z^{-3} + \bar{a}_2 z^{-2} + \bar{a}_1 z^{-1} + a_0 + a_1 z + a_2 z^2 + a_3 z^3.$$

Let a linear projection operator T_m be defined. For $G \in \mathbb{C}^{N\times M}$, $T_m(G)$ is found by summing the elements in each diagonal of G and placing the result in the upper left element of the diagonal, which is in either the first column or first row. The remaining entries in the diagonal are set equal to zero. For any matrix G, denote the structured matrix $T_m(G)$ as a *bordered matrix.* For example,

$$T_m\left(\begin{bmatrix} a & b & c \\ e & f & g \\ i & j & k \\ m & n & o \end{bmatrix}\right) = \begin{bmatrix} a+f+k & b+g & c \\ e+j+o & 0 & 0 \\ i+n & 0 & 0 \\ m & 0 & 0 \end{bmatrix}.$$

It follows from (1) and (2) and the definition of T_m that the coefficients a_k of the rational function $r(z) = v(\omega)^H c\ c^H v(\omega)$ are equal to the elements on the border of the matrix C. The following result is therefore obtained.

Theorem 1 *Let $A_1, A_2 \in \mathbb{C}^{N\times N}$. Then*

$$\begin{aligned} T_m(A_1^H A_1) &= T_m(A_2^H A_2) \\ &\Downarrow \\ v(\omega)^H A_1^H A_1 v(\omega) &= v(\omega)^H A_2^H A_2 v(\omega). \end{aligned}$$

If c is an $N-$vector then $T_m(cc^H) \in \mathbb{C}^{N\times N}$ is in general a rank-two Hermitian matrix due to its structure, with the eigendecomposition

$$T_m(cc^H) = \gamma yy^H + \beta ww^H.$$

with

$$y = \begin{bmatrix} y_1 \\ \tilde{y} \end{bmatrix} \quad \text{and} \quad w = \begin{bmatrix} w_1 \\ \tilde{w} \end{bmatrix}.$$

where y_1 and w_1 are scalars, $\tilde{y}$ and $\tilde{w}$ are $(N-1)$-vectors and where

$$\begin{aligned} T_m(cc^H)\,[2:N, 2:N] &= \gamma\tilde{y}\tilde{y}^H + \beta\tilde{w}\tilde{w}^H \\ &= \begin{bmatrix} 0 & \cdots & 0 \\ & \cdots & \\ 0 & \cdots & 0 \end{bmatrix}. \end{aligned}$$

In particular,

$$\gamma|\tilde{y}_i|^2 + \beta|\tilde{w}_i|^2 = 0, \quad i = 1, \ldots, N-1. \qquad (3)$$

The eigenvalues γ and β are real since $T_m(cc^H)$ is Hermitian and it follows from (3) that γ and β are of opposite sign. Since A^HA has real nonnegative eigenvalues whereas the eigenvalues of $T_m(cc^H)$ are of opposite sign, it follows that nontrivial matrices A such that $A^HA = T_m(cc^H)$ do not exist. Fortunately this is not necessary, because it is $T_m(A^HA)$ which determines the magnitude response of the filter, A, and not A^HA. The first design goal then was to find a matrix, A such that:

$$T_m(A^HA) = T_m(cc^H).$$

3. IMPLICATIONS OF MATRIX FILTER RANK.

The implications involving the rank of the matrix filter, A will now be considered. The linear convolution $b * v$ where $b \in \mathbb{C}^M$ and $v \in \mathbb{C}^N$ can be written in matrix format as

$$x = b * v = Bv. \qquad (4)$$

where $B \in \mathbb{C}^{(N+M-1)\times N}$ is a convolution matrix. It follows that $x \in \mathbb{C}^{N+M-1}$. The determination of v given B and x in (4) with $M > 1$ involves an overdetermined set of linear equations which in general cannot be solved exactly. If B has full column rank, however, then the original vector, v can be recovered by solving the normal equations $v = (B^HB)^{-1}B^Hx$ because x lies in the column space of B. It follows that when B is full column rank, (1) distinct data vectors are mapped into distinct filtered vectors, (2) the filtered vector is nonzero whenever the data vector is and (3) a data vector can be recovered from its filtered version.

Recall that the *kernel* or *nullspace*, $N(G)$ associated with a matrix, $G \in \mathbb{C}^{N+M-1\times N}$ is a subspace of $\mathbb{C}^N$ consisting of the set of elements, y such that $Gy = 0$. If S is a subspace in $\mathbb{C}^N$ then $S^\perp$ is the subspace consisting of all elements, $z \in \mathbb{C}^N$ such that $z^Hy = 0$, for all $y \in S$. Consider a matrix filter, G which is rank deficient. Then (1) any two distinct data vectors differing by a data vector in the kernel $N(G)$ of G will have the same filter output, (2) any data vector in $N(G)$ will have the filter output zero and (3) any data vector, w with decomposition $w = x + y,\ x \in N(G)^\perp,\ y \in N(G),\ y \neq 0$ cannot be recovered once the filtering operation has taken place. It goes without saying that these are undesirable properties which are not shared by any linear convolution filter which has a full rank convolution matrix. The second design goal in addition to matching magnitude responses was therefore to design a full rank matrix filter.

4. DEVELOPMENT OF THE NONLINEAR EQUATIONS.

One candidate matrix filter is a rank one modification of a scaled identity matrix:

$$A = xx^H + \alpha I. \qquad (5)$$

The matrix A is nonsingular as long as $\alpha \neq -x^Hx$; it will be assumed that this is true in what follows. The filtering operation, $Av = (xx^H + \alpha I)v = x(x^Hv) + \alpha v$, $v \in \mathbb{C}^N$ takes $10N$ real flops when A is real. The Sherman-Morrison-Woodbury formula [2] can be used to write the inverse filter A^{-1} as a rank-one update to the scaled identity:

$$\left(xx^H + \alpha I\right)^{-1} = \alpha^{-1}I - \left(\frac{\alpha^{-2}}{1 + x^Hx/\alpha}\right)xx^H.$$

Using (5) the following is obtained:

$$\begin{aligned} A^HA &= (xx^H + \alpha I)^H(xx^H + \alpha I) \\ &= (2\mathrm{Re}(\alpha) + x^Hx)xx^H + |\alpha|^2 I. \end{aligned}$$

Let f be a given FIR filter and let $c = [c_0\, c_1\, \cdots\, c_{N-1}]^T$. Define $H = T_m(cc^H)$ and let $M = T_m(A^HA)$. Let $H[:, n]$ denote the nth column of the matrix, H. Then the system of equations to be solved is:

$$M[:, 1] - H[:, 1] = 0.$$

The first columns of the matrices M and H are thus required to be equal. Let $s = H[:, 1]$. For real M, H and $N = 4$ these equations are:

$$\begin{aligned} y(x_1^2 + x_2^2 + x_3^2 + x_4^2) + 4\alpha^2 - s_0 &= 0. \\ y(x_1x_2 + x_2x_3 + x_3x_4) - s_1 &= 0. \\ y(x_1x_3 + x_2x_4) - s_2 &= 0. \\ y(x_1x_4) - s_3 &= 0. \qquad (6) \end{aligned}$$

where $y = (2\alpha + x_1^2 + x_2^2 + x_3^2 + x_4^2)$. The existence of solutions to (6) has been verified by using the computer algebra program, MAPLE [3]. In order to obtain an increased degree of freedom the scalar, α, was treated as an unknown to be solved for along with the elements of the vector, x, by adding an additional independent equation. The set of $N + 1$ equations passed to the nonlinear equation solver was:

$$\begin{aligned} \|M - H\|_F &= 0. \quad \text{(1 equation)}. \\ M[:, 1] - H[:, 1] &= 0. \quad \text{(N equations)}. \qquad (7) \end{aligned}$$

In order to solve (7) a quasi-Newton method for the solution of nonlinear systems of equations called Broyden's method [4, 5] was used because explicit formulas for the elements in the Jacobian matrix are not

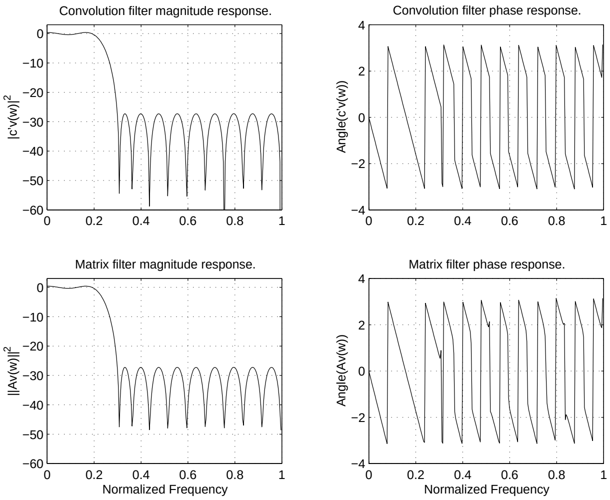

Fig. 1. Magnitude and Phase Responses

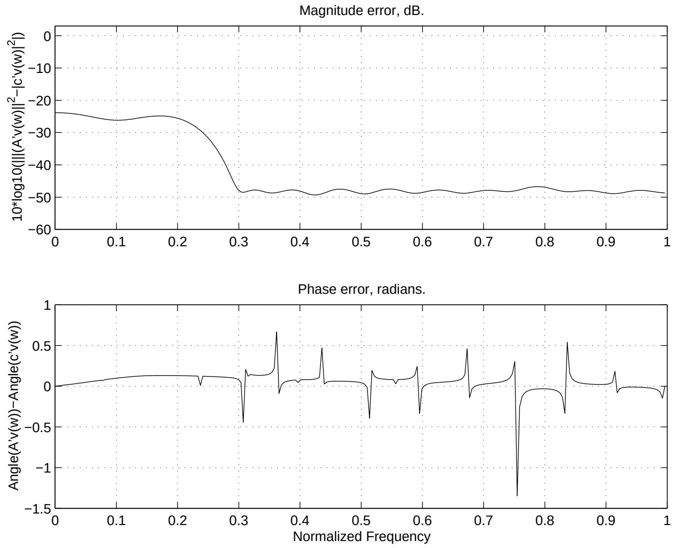

Fig. 2. Matrix Filter Magnitude and Phase Errors.

required. Real matrix equiripple filters up to size 40 have been designed using this method and in all cases acceptable solutions have been found, although starting Broyden's method with several different initial values for x and α was sometimes necessary. However, this does not imply that matrix filter counterparts to arbitrary equiripple FIR filters always exist.

5. AN EQUIRIPPLE MATRIX FILTER.

The following example illustrates the design of a 26×26 matrix filter. An equiripple FIR filter was designed in MATLAB, using the following command:

```
pfilt = remez(25,[0 .2 .3 1],[1 1 0 0]);
```

The coefficient vector pfilt is real and symmetric, so that the vector c satisfies:

$$[c_0 \;\cdots\; c_{N-1}]^T = [\bar{c}_{N-1} \;\cdots\; \bar{c}_0]^T.$$

The procedure once the filter coefficients were found was to solve the system of nonlinear equations (7) numerically, starting with approximate values for $x \in \mathbb{R}^{26}$ and $\alpha \in \mathbb{R}$. The MATLAB plot in Figure 1 gives magnitude and phase responses for the FIR equiripple filter and matrix filter. Figure 2 shows the corresponding magnitude squared error in dB and the phase error in radians. The RMS magnitude error was 0.0015.

6. CONCLUSIONS

It has been demonstrated that it is possible to design matrix filters whose frequency responses closely approximate the responses of linear equiripple FIR filters. There are many matters remaining to be investigated and understood. For example, whether all equiripple FIR filters can be implemented as matrix filters using the method in this paper depends on the existence, uniqueness and computability of solutions to the system of equations in (7). In addition, a proof of the existence of matrix filters which improve upon the performance of equiripple FIR filters remains to be given.

7. REFERENCES

[1] R. Vaccaro, B. Harrison, Optimal Matrix Filter Design, IEEE Transactions on Signal Processing 44 (3) (1996) 705–709.

[2] G. Golub, C. V. Loan, Matrix Computations, 3rd Edition, Johns Hopkins University Press, 1996.

[3] A. Heck, Introduction to MAPLE, Springer Verlag, 1996.

[4] W. Press, B. Flannery, S. Teukolsky, W. Vetterling, Numerical Recipes, The Art of Scientific Computing, Cambridge University Press, 1990.

[5] J. J.E. Dennis, R. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, SIAM, 1996.