Reprint

J.M. Costa and A.N. Venetsanopoulos, "Recursive implementation of factorable two-dimensional digital filters", *Canadian Electrical Engineering Journal*, Vol. 4, No. 3, pp. 33-40, July 1979.

Recursive implementation of factorable two-dimensional digital filters

By J. M. Costa, *Bell Northern Research, Ottawa, and* A. N. Venetsanopoulos, *University of Toronto.*

A brief tutorial background on the design of two-dimensional digital filters is presented. A technique for designing stable two-dimensional recursive filters is summarized and an algorithm useful for controlling the cutoff frequencies is discussed. The problem of implementation of two-dimensional infinite impulse-response digital filters is then considered and an algorithm is proposed for filters which are factorable into bilinear second-order sections with complex coefficients. Computational errors due to the finite wordlength of the filters are analyzed. These errors can be represented by a set of error sequences additively contaminating the output of an ideal realization. The transfer functions relating these error sequences to simple roundoff errors are derived. The problems of data management when large matrices are stored in auxiliary storage are also discussed and a solution is proposed.

Cet article contient un bref rappel sur la conception des filtres numériques à deux dimensions. On trouvera un résumé d'une technique de fabrication de filtres récurrents à deux dimensions stables et une discussion à propos d'un algorithme utile dans la maitrise des fréquences de coupures. L'auteur envisage ensuite le problème de la mise en place de filtres numériques à réponse d'impulsion, infinis à deux dimensions et propose un algorithme pour les filtres qu'il est possible de mettre en facteur en sections du deuxième ordre bilinéaires affectées de coefficients complexes. Il analyse également les erreurs de calculs dues à la longueur de mot finie des filtres. On peut représenter ces erreurs par un jeu de séquences d'erreurs contaminant de façon cumulative la sortie d'une réalisation idéale. On a dérivé les fonctions de transfert liant ces séquences d'erreur aux simples erreurs d'arrondissement. L'article se termine par une discussion sur le problème de la gestion des données quand des matrices importantes sont stockées dans des mémoires auxiliaires et par une proposition de solution à ce problème.

Introduction

The digital filtering of two-dimensional signals offers the many advantages characteristic of digital computers, such as flexibility and accuracy. Applications exist in the processing of images and geophysical data.

Two-dimensional digital filter design techniques can be divided into two groups, those for finite impulse response (FIR) filters and those for infinite impulse response (IIR) filters. Comprehensive reviews of these techniques have been presented by Mersereau and Dudgeon[1] and Chakrabarti and Mitra[2]. Recursive realizations have the potential of saving both computer time and memory.[3] It is well known that high-order two-dimensional IIR filters can not be, in general, factored into lower order filters and realized in parallel or cascade form to reduce the effect of computational noise.[4] Filtering algorithms for two-dimensional recursive filters have been given by Shanks.[5] They can be implemented using either the direct form or a state-variable description of the filter. Other algorithms using direct forms and continued fraction expansions have been proposed by Mitra, Sagar, and Pendergrass.[6]

This paper summarizes a design technique previously used for two-dimensional IIR filters,[7] introduces an algorithm useful for controlling the cutoff frequencies and considers the problem of implementation of two-dimensional IIR filters, which are factorable into bilinear second-order sections with complex coefficients.

The filter design and realization techniques described in this paper have been applied by Harrison to the processing of geophysical data for the point determinations of ore grade from a Canadian copper mine.[8]

Design of IIR filters

One of the difficulties for the design of IIR filters in two-dimensions is due to the fact that a polynomial in two variables $P(z_1, z_2)$ cannot in general be factored into first- or second-order polynomials. This implies that many one-dimensional design techniques cannot be readily extended to two-dimensions and that a high-order two-dimensional filter cannot in general be synthesized in parallel or cascade form to reduce the effect of quantization noise. It is also difficult to test the stability of two-dimensional IIR filters except in simple cases. Stability is an important issue in the design of IIR filters.[1] If a filter is unstable, any input, including computational noise, could cause the output to grow without bound, obliterating the desired response.

A two-dimensional recursive filter can be described by a linear difference equation. The general form is

$$g(m,n) = \sum_{i=1}^{M_b}\sum_{j=1}^{N_b} a_{ij} f(m-i+1, n-j+1) - \sum_{\substack{i=1\\ i,j\neq 1 \text{ simultaneously}}}^{M_b}\sum_{j=1}^{N_b} b_{ij}\, g(m-i+1, n-j+1), \tag{1}$$

Here it is assumed either that all output values $g(m-i+1, n-j+1)$ have been computed previously or are equal to zero (boundary conditions).

Equation (1) can be written in the form

$$\sum_{i=1}^{M_b}\sum_{j=1}^{N_b} b_{ij}\, g(m-i+1, n-j+1) = \sum_{i=1}^{M_a}\sum_{j=1}^{N_a} a_{ij} f(m-i+1, n-j+1) \tag{2}$$

where $b_{11} = 1$.

Equation (2) can be solved for $g(m,n)$ [cf. (1)], $g(m-M_b+1, n)$, $g(m, n-N_b+1)$, or $g(m-M_b+1, n-N_b+1)$ and in each case the difference equation obtained corresponds to a two-dimensional recursive filter recursing in the $(+m, +n)$, $(-m, +n)$, $(+m, -n)$, or $(-m, -n)$ directions, respectively.[9]

A useful representation of (2) is obtained by the two-dimensional z-transform or double z-transform. This transform is defined as follows:

$$X(z_1, z_2) \triangleq \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} x(m,n)\, z_1^m z_2^n \tag{3}$$

where $x(m,n)$ is a two-dimensional sequence, z_1 and z_2 are delay operators, and $X(z_1, z_2)$ its two-dimensional z-transform. The domain

This work was supported by the National Research Council of Canada under Grant No. A-7397.

of definition for $X(z_1,z_2)$ is its region of convergence in the z_1,z_2 plane. The properties of the two-dimensional z-transform are similar to those of the z-transform in one-dimension.

If the sequence $x(m,n)$ is of finite duration and is bounded, then (3) has finite limits in all the summations and $X(z_1,z_2)$ converges everywhere in the z_1z_2 plane.

The two-dimensional z-transform of (2) can be shown[1] to be

$$A(z_1,z_2)F(z_1,z_2) = B(z_1,z_2)G(z_1,z_2)$$

or

$$G(z_1,z_2) = H(z_1,z_2)F(z_1,z_2)$$

where $F(z_1,z_2)$ and $G(z_1,z_2)$ are the two-dimensional z-transforms of the input and the output sequence of the filter, respectively, and

$$H(z_1,z_2) = \frac{A(z_1,z_2)}{B(z_1,z_2)} \triangleq \frac{\sum_{i=1}^{M_a}\sum_{j=1}^{N_a} a_{ij} z_1^{i-1} z_2^{j-1}}{\sum_{i=1}^{M_b}\sum_{j=1}^{N_b} b_{ij} z_1^{i-1} z_2^{j-1}} \tag{4}$$

is the transfer function of the filter. The frequency response of the filter is obtained by evaluating (4) for $z_1 = \exp(-j2\pi f_1T_1)$ and $z_2 = \exp(-j2\pi f_2T_2)$, where T_1 and T_2 are the sample intervals in the m and n directions, respectively.

Frequency domain design techniques for two-dimensional IIR filters can be divided into optimum design techniques and ad hoc design techniques.

Optimum filters may be more efficient in terms of the realization but the computation time of the design can become prohibitively large.

The ad hoc design techniques proposed so far consist of cascading elementary one- and two-dimensional filters. A class of elementary filters can be obtained with a design method due to Shanks,[10, 11] which maps one-dimensional into two-dimensional filters with arbitrary directivity in the two-dimensional frequency response plane. These filters are called rotated filters, because they are obtained by rotating one-dimensional filters in the two-dimensional frequency response plane. Indeed, given the transfer function of an analogue filter H(s) (e.g. Butterworth, Gaussian, etc.) the application of the transformation

$$s = -s_1 \sin\theta + s_2 \cos\theta$$

results in a transfer function whose frequency response is the original one but rotated by the angle θ. The corresponding two-dimensional recursive filter is obtained by applying the bilinear transformation to each variable s_1 and s_2. This transformation distorts the frequency response especially at frequencies close to the Nyquist frequency, consequently the magnitude response has the shape of a ridge. The angle of rotation θ is defined as that extended by the direction of the crest (cf. Ref. 7, Fig. 2). The cutoff frequency of a rotated filter is then usually measured in a direction $\theta + 90°$, namely on a cross-cut perpendicular to the crest and passing through the origin.

Rotated filters are stable if the angle of rotation θ lies between 270° and 370°. In certain applications a slight perturbation of the coefficients may be necessary to ensure stability for all inputs, as indicated in Ref. 7. A rigorous discussion of the stability of rotated filters has been presented by Goodman.[12, Appendix I] To implement filters with angles of rotation outside the range $270° < \theta < 360°$ a data transformation technique was proposed.[7, 13, 14] Its essence consists in transforming both the data and the transfer function of the filter to guarantee stability. The transformations of the data do not affect the filter stability and the output can be inverse transformed, so that the overall transfer function is the desired one. Indeed, it is easily shown that the system

$$Y(w_1,w_2) = [T\,H(w_1,w_2)] \cdot X(w_1,w_2)$$

is equivalent to the system

$$Y(w_1,w_2) = T\{H(w_1,w_2) \cdot [T^{-1}X(w_1,w_2)]\}$$

where $X(w_1,w_2)$ and $Y(w_1,w_2)$ are the Fourier transforms of the input and output of the system, respectively, and the transform function of the system is $H(w_1,w_2)$ affected by the transformation T.[13]

There are eight possible transformations, as shown in Ref. 13. Figure 1 (cf. Fig. 2 in Ref. 13) shows the direction of recursion, sense of recursion, and the starting point of the input data for each of the transformed filters. A horizontal arrow denotes a recursion by rows and a vertical arrow denotes a recursion by columns. The head of the arrow points in the sense of recursion and the base of the arrow shows the starting point. Figure 1 will prove to be crucial in understanding the filter implementation technique presented under "Realization of two-dimensional IIR filters in a general purpose digital computer". An alternative to these transformations is to filter the data in a different manner, by using an algorithm recursing in another direction.[11, 13, 14]

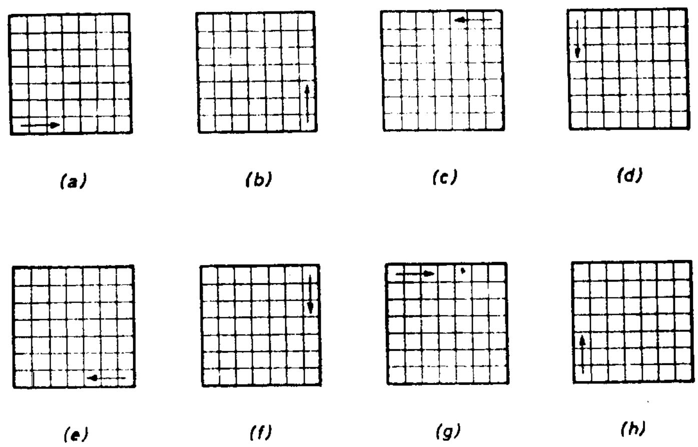

Figure 1 (a)-(h): *Realization of a transformed filter by changing the direction and sense of recursion of the filter algorithm.*

Hence, stable rotated filters can be obtained for any effective angle of rotation. In a previous paper we used this method to design good approximations to circularly-symmetric filters by cascading rotated filters.[7] To approximate the circular symmetry the angles of rotation are chosen uniformly distributed over 180° (or 360° if zero-phase response is required). An extension to the design method in Ref. 7 which facilitates frequency response optimization via nonlinear programming has recently been published by Goodman.[12] In the next section we present an iterative technique for controlling the cutoff frequencies of a cascade of rotated filters.

Control of the cutoff frequencies

Rotated filters are designed in the s-domain and the two-dimensional bilinear z-transforms are used to obtain digital filters. Due to the double periodicity of the frequency response of two-dimensional digital fiters, the desired frequency response of rotated filters is distorted to an extent which depends on the angle of rotation. We have observed that in the low-frequency region the deviation from the desired frequency response is maximum for a rotation of 315°. The previously mentioned algorithm for designing circularly symmetric filters[7] did not compensate for this effect. The frequency response was approximated iteratively in one direction only and each rotated filter in the cascade was designed with the same cutoff frequency.

We have developed[15] a new algorithm in which the frequency response is approximated in as many directions as the number of rotated filters being cascaded. The cutoff frequency of each rotated filter is adjusted separately and iteratively according to the frequency response obtained when all the filters in the cascade are considered.

The approach has not only the advantage of giving a better circularly-symmetric frequency response but also non-circularly-symmetric filters can be designed by specifying different cutoff frequencies in each direction. Since only low-pass filters are cascaded, the locus of the cutoff frequency of the two-dimensional filter cannot be arbitrary and consequently some constraints are imposed on the relationships among the cutoff frequencies. Band-pass and high-pass filters can be obtained by combining low-pass and all-pass filters in parallel.

Using this new algorithm problems of convergence may arise if too much accuracy is specified in too many directions. Indeed, the adjustment of the cutoff frequency of each rotated filter in the cascade affects the frequency response over the whole plane, thus also changing the cutoff frequencies in the other directions, in a manner which may conflict with the desired specifications. The algorithm takes care of this problem by stopping the iteration as soon as it ceases to converge. An error code is returned, which specifies if convergence was attained. The source listing of a FORTRAN IV Program which implements this algorithm is given in Ref. 16.

The new algorithm features the following steps:

1. Initialization. Get the poles and zeros of a stable one-dimensional continuous low-pass filter with cutoff frequency normalized to $1/(2\pi)$, (i.e. cutoff angular frequency normalized to unity). Get the K_θ desired angles of rotation $270° \leq \theta(k) \leq 360°$, $k = 1, \ldots K_\theta$ and the desired cutoff frequencies $f_u(k)$, $k = 1, \ldots, K_\theta$ of the two-dimensional digital filter in each direction $\theta(k) + 90°$, $k = 1, \ldots, K_\theta$. The frequencies $f_u(k)$ are given as fractions of half the sampling frequency. If the input arguments are inconsistent, return to the calling program with an error code IERROR = 1. Otherwise let $i = 0$ for the first iteration and continue.
2. Let $f_d^i(k) = f_u(k)$, $k = 1, \ldots, K_\theta$.
3. For $k = 1, \ldots, K_\theta$ determine the coefficients of the K_θ rotated digital filters derived from the one-dimensional continuous filter by multiplying the poles by $(\pi/2)f_d^i(k)$ or by $\tan[(\pi/2)f_d^i(k)]$ if the frequency axis is to be prewarped (cf. (11) in Ref. 7).
4. With the K_θ rotated filters in cascade find the cutoff frequencies of the resultant two-dimensional filter, $f_c^i(k)$, $k = 1, \ldots, K_\theta$, in the directions $\theta(k) + 90°$, $k = 1, \ldots, K_\theta$, respectively. If the algorithm that searches for these cutoff frequencies does not converge, then return to the calling program with IERROR = 3.
5. If $|f_u(k) - f_c^i(k)| < \varepsilon$ for $k = 1, \ldots, K_\theta$ where ε is the specified maximum error of the cutoff frequency in each direction; then return to the calling program with IERROR = 0. In this case the execution is completed satisfactorily.
6. If there is no improvement in the last iteration step, that is if

$$\sum_{k=1}^{K_\theta} |f_u(k) - f_c^i(k)| \geq \sum_{k=1}^{K_\theta} |f_u(k) - f_c^{i-1}(k)|$$

then return with IERROR = 4.

7. Let $i \leftarrow i+1$. If i is greater than the maximum allowed number of iterations then return with IERROR = 2.
8. Let $f_d^i(k) = f_d^{i-1}(k) + f_u(k) - f_c^{i-1}(k)$, $k = 1, \ldots, K_\theta$.
9. Go back to step 3.

Since stable rotated filters can be obtained only for angles of rotation between 270° and 360°, a parameter is used which specifies how the filter is to be combined with data transformations to obtain the desired symmetries. Another parameter (normally set to zero) may be used to modify the coefficients slightly to guarantee stability for all inputs (cf. Fig. 3 in Ref. 7). Thus, the overall system has the required cutoff frequencies and is stable.

For non-circularly-symmetric filters, it may seem possible to obtain elliptic shapes or other shape types from circularly-symmetric filters by changing the scaling along each axis. However, in general, this is not possible because the filter would most probably become unstable. On the other hand, the technique that we described previously results in stable filters.

Two-dimensional complex cascade programming

Since the previous design technique leads to a factorable transfer function an algorithm is now proposed for the realization of two-dimensional IIR filters whose transfer function is factorable as follows:

$$H(z_1,z_2) = A\prod_{i=1}^{M}\frac{a_{11}^i + a_{21}^i z_1 + a_{12}^i z_2 + a_{22}^i z_1 z_2}{b_{11}^i + b_{21}^i z_1 + b_{12}^i z_2 + b_{22}^i z_1 z_2} \qquad (5)$$

where A is a real gain constant and the coefficients a_{jk}^i and b_{jk}^i, $(j,k = 1,2)$ may be complex. In this paper it is assumed that their conjugates a_{jk}^{i*} and b_{jk}^{i*} $(j,k = 1,2)$ are also present in the corresponding product, to result in a filter impulse response which is real. Common factors have been removed from the numerator and the denominator.

Equation (5) can be physically interpreted as a cascade of bilinear second-order systems and can be readily realized by complex cascade programming. The technique of one-dimensional complex cascade programming was introduced by Knowles and Edwards[7] and enables a pulsed transfer function with complex poles and zeros to be realized. The operations are programmed using real arithmetic and are simplified using the fact that the impulse response of each pair of conjugate filters is real. The cascade form of two-dimensional recursive filters has some advantages over the direct and parallel forms and results in a reduction of the output computational error.[17,18]

For complex cascade programming (5) is expressed in the form*

$$H(z_1,z_2) = A\prod_{i=1}^{N_R} H_i^R(z_1,z_2) \prod_{j=N_R+1}^{N} H_j(z_1,z_2)H_j^*(z_1^*,z_2^*) \qquad (6)$$

where $2N - N_R = M$, $H_i^R(z_1,z_2)$ for $0 < i \leq N_R$ are the filters in the cascade which have real coefficients, and $H_j(z_1,z_2)$ and $H_j^*(z_1^*,z_2^*)$ for $N_R < j \leq N$ are the remaining complex filters in the cascade. That is

$$H_i^R(z_1,z_2) \triangleq \frac{1 + a_{21}^i z_1 + a_{12}^i z_2 + a_{22}^i z_1 z_2}{1 + b_{21}^i z_1 + b_{12}^i z_2 + b_{22}^i z_1 z_2} \qquad 0 < i \leq N_R \qquad (7a)$$

where the a's and b's are real.

$$H_j(z_1,z_2) \triangleq \frac{1 + a_{21}^j z_1 + a_{12}^j z_2 + a_{22}^j z_1 z_2}{1 + b_{21}^j z_1 + b_{12}^j z_2 + b_{22}^j z_1 z_2} \qquad N_R < j \leq N \qquad (7b)$$

$$H_j^*(z_1^*,z_2^*) \triangleq \frac{1 + a_{21}^{j*} z_1 + a_{12}^{j*} z_2 + a_{22}^{j*} z_1 z_2}{1 + b_{21}^{j*} z_1 + b_{12}^{j*} z_2 + b_{22}^{j*} z_1 z_2} \qquad N_R < j \leq N \qquad (7c)$$

where the a's and b's are complex and the a*'s and b*'s are their complex conjugates. Without loss of generality we can assume that $a_{11}^i = b_{11}^i = 1$ for $0 < i \leq N$, by appropriate adjustment of the scalar gain factor A.

A block diagram for complex cascade programming is given in Fig. 2. With reference to this block diagram and equations (7), the complex cascade programming of $H(z_1,z_2)$ consists of successively solving a set of difference equations for $g(k,\ell)$ where k and ℓ are the two-dimensional sequence indices in the directions which correspond to the unit delay operators z_1 and z_2, respectively. These equations are shown in Table 1.

Equation (8b) corresponds to the realization of the transfer function in (7a). Equations (8c) and (8d) correspond to the transfer function in (7b), while (8e) corresponds to the transfer function in (7c). It should be observed that, for each pair (k,ℓ), the sequence $\{d_i(k,\ell): i = 1,2,\ldots,N\}$ is real and the sequence $\{e_i(k,\ell): i = N_R+1, N_R+2, \ldots, N\}$ is complex (see Fig. 2). These two sequences are simply intermediate values towards the calculation of the final output $g(k,\ell)$, when the input is $f(k,\ell)$. A feature of the recursive equations (8) is that they involve real arithmetic operations only.

The notation $H_j^*(z_1^*,z_2^*)$ may seem a bit awkward but it is mathematically necessary to denote that only the coefficients of $H_j(z_1,z_2)$ are conjugated, not the variables z_1 and z_2, which are affected by two complex conjugate star signs that cancel.

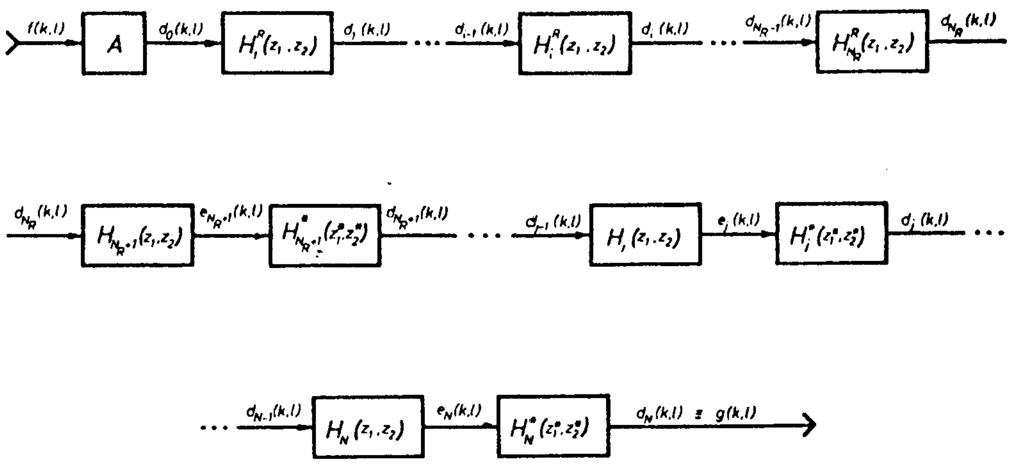

Figure 2: *Block diagram for two-dimensional complex cascade programming.*

TABLE 1

$$d_0(k,\ell) = Af(k,\ell) \tag{8a}$$

$$\begin{aligned} d_i(k,\ell) = {} & d_{i-1}(k,\ell) + a^i_{21}d_{i-1}(k-1,\ell) + a^i_{12}d_{i-1}(k,\ell-1) + a^i_{22}d_{i-1}(k-1,\ell-1) + \\ & - b^i_{21}d_i(k-1,\ell) - b^i_{12}d_i(k,\ell-1) - b^i_{22}d_i(k-1,\ell-1) \qquad \text{for } i = 1,2,\ldots,N_R \end{aligned} \tag{8b}$$

$$\begin{aligned} \mathrm{Re}[e_i(k,\ell)] = {} & d_{i-1}(k,\ell) + \mathrm{Re}[a^i_{21}]d_{i-1}(k-1,\ell) + \mathrm{Re}[a^i_{12}]d_{i-1}(k,\ell-1) + \\ & + \mathrm{Re}[a^i_{22}]d_{i-1}(k-1,\ell-1) - \mathrm{Re}[b^i_{21}]\mathrm{Re}[e_i(k-1,\ell)] + \mathrm{Im}[b^i_{21}]\mathrm{Im}[e_i(k-1,\ell)] + \\ & - \mathrm{Re}[b^i_{12}]\mathrm{Re}[e_i(k,\ell-1)] + \mathrm{Im}[b^i_{12}]\mathrm{Im}[e_i(k,\ell-1)] - \mathrm{Re}[b^i_{22}]\mathrm{Re}[e_i(k-1,\ell-1)] + \\ & + \mathrm{Im}[b^i_{22}]\mathrm{Im}[e_i(k-1,\ell-1)] \qquad \text{for } i = N_R+1,\ldots,N \end{aligned} \tag{8c}$$

$$\begin{aligned} \mathrm{Im}[e_i(k,\ell)] = {} & \mathrm{Im}[a^i_{21}]d_{i-1}(k-1,\ell) + \mathrm{Im}[a^i_{12}]d_{i-1}(k,\ell-1) + \mathrm{Im}[a^i_{22}]d_{i-1}(k-1,\ell-1) + \\ & - \mathrm{Im}[b^i_{21}]\mathrm{Re}[e_i(k-1,\ell)] - \mathrm{Re}[b^i_{21}]\mathrm{Im}[e_i(k-1,\ell)] - \mathrm{Im}[b^i_{12}]\mathrm{Re}[e_i(k,\ell-1)] + \\ & - \mathrm{Re}[b^i_{12}]\mathrm{Im}[e_i(k,\ell-1)] - \mathrm{Im}[b^i_{22}]\mathrm{Re}[e_i(k-1,\ell-1)] + \\ & - \mathrm{Re}[b^i_{22}]\mathrm{Im}[e_i(k-1,\ell-1)] \qquad \text{for } i = N_R+1,\ldots,N \end{aligned} \tag{8d}$$

$$\begin{aligned} d_i(k,\ell) = {} & \mathrm{Re}[e_i(k,\ell)] + \mathrm{Im}[a^i_{11}]\mathrm{Im}[e_i(k,\ell)] + \mathrm{Re}[a^i_{21}]\mathrm{Re}[e_i(k-1,\ell)] + \\ & + \mathrm{Im}[a^i_{21}]\mathrm{Im}[e_i(k-1,\ell)] + \mathrm{Re}[a^i_{12}]\mathrm{Re}[e_i(k,\ell-1)] + \mathrm{Im}[a^i_{12}]\mathrm{Im}[e_i(k,\ell-1)] + \\ & + \mathrm{Re}[a^i_{22}]\mathrm{Re}[e_i(k-1,\ell-1)] + \mathrm{Im}[a^i_{22}]\mathrm{Im}[e_i(k-1,\ell-1)] - \mathrm{Re}[b^i_{21}]d_i(k-1,\ell) + \\ & - \mathrm{Re}[b^i_{12}]d_i(k,\ell-1) - \mathrm{Re}[b^i_{22}]d_i(k-1,\ell-1) \qquad \text{for } i = N_R+1,\ldots,N \end{aligned} \tag{8e}$$

$$g(k,\ell) = d_N(k,\ell) \tag{8f}$$

It should be noted that the number of real multiples required for the realization of each complex pair of filters is higher than those required if each pair was replaced by a fourth order section with real coefficients. However, this method still has advantages over the direct method. The number of multiples is not the only criterion for optimum filter realization. Stability characteristics, coefficient sensitivity and computational noise are generally worse in higher order sections. Other considerations, such as the number of delays and the form of the transfer function, given by the design technique, are also important. Indeed, when the filter design technique leads to a transfer function of the form given by (5), the complex cascade programming realization allows the immediate realization, without any further manipulation of coefficients, which would require additional operations. This was one of the reasons which led to our investigation of complex cascade realizations in two dimensions.

Computational errors in two-dimensional complex cascade programming

Since actual operations are carried out with only a finite number of bits there is a cumulative roundoff error which propagates through the stages of the filter. In the steady state, the computational error can be represented by a set of additive noise processes at the output of an infinite precision arithmetic filter.[19] We shall determine an expression for the computational error to be added to the output of the ideal realization to account for the finite wordlength of the actual filter. This analysis is an extension of the one-dimensional analysis of Ref. 17.

If we call $\{w_i(k,\ell): i = 0,1,2,\ldots,3N-2N_R\}$, the roundoff errors incurred in the evaluation of equations (8), then the actual outputs of the elements in Fig. 2 become the quantities shown in Table 2.

Defining the computational error quantities as the differences between the actual and the ideal outputs of the elements in Fig. 2, we have

$$\varepsilon_0(k,\ell) \triangleq d_0'(k,\ell) - d_0(k,\ell) \tag{10a}$$

$$\varepsilon_i(k,\ell) \triangleq d_i'(k,\ell) - d_i(k,\ell) \qquad 0 < i \leq N_R \tag{10b}$$

$$\varepsilon_{3i-2N_R-2}(k,\ell) \triangleq \mathrm{Re}[e_i'(k,\ell)] - \mathrm{Re}[e_i(k,\ell)] \qquad N_R < i \leq N \tag{10c}$$

$$\varepsilon_{3i-2N_R-1}(k,\ell) \triangleq \mathrm{Im}[e_i'(k,\ell)] - \mathrm{Im}[e_i(k,\ell)] \qquad N_R < i \leq N \tag{10d}$$

$$\varepsilon_{3i-2N_R}(k,\ell) = d_i'(k,\ell) - d_i(k,\ell) \qquad N_R < i \leq N \tag{10e}$$

then from (8f), (9f) and (10e) the computational error $\varepsilon_c(k,\ell)$ at the filter output is

$$\varepsilon_c(k,\ell) = g'(k,\ell) - g(k,\ell)$$

TABLE 2

$$d'_0(k,\ell) = Af(k,\ell) + w_0(k,\ell) \tag{9a}$$

$$\begin{aligned} d'_i(k,\ell) = {} & d'_{i-1}(k,\ell) + a^i_{21}d'_{i-1}(k-1,\ell) + a^i_{12}d'_{i-1}(k,\ell-1) + a^i_{22}d'_{i-1}(k-1,\ell-1) + \\ & - b^i_{21}d'_i(k-1,\ell) - b^i_{12}d'_i(k,\ell-1) - b^i_{22}d'_i(k-1,\ell-1) + w_i(k,\ell) \qquad \text{for } i = 1,2,\ldots,N_R \end{aligned} \tag{9b}$$

$$\begin{aligned} \mathrm{Re}[e'_i(k,\ell)] = {} & d'_{i-1}(k,\ell) + \mathrm{Re}[a^i_{21}]d'_{i-1}(k-1,\ell) + \mathrm{Re}[a^i_{12}]d'_{i-1}(k,\ell-1) + \\ & + \mathrm{Re}[a^i_{22}]d'_{i-1}(k-1,\ell-1) - \mathrm{Re}[b^i_{21}]\mathrm{Re}[e'_i(k-1,\ell)] + \mathrm{Im}[b^i_{21}]\mathrm{Im}[e'_i(k-1,\ell)] + \\ & - \mathrm{Re}[b^i_{12}]\mathrm{Re}[e'_i(k,\ell-1)] + \mathrm{Im}[b^i_{12}]\mathrm{Im}[e'_i(k,\ell-1)] - \mathrm{Re}[b^i_{22}]\mathrm{Re}[e'_i(k-1,\ell-1)] + \\ & + \mathrm{Im}[b^i_{22}]\mathrm{Im}[e'_i(k-1,\ell-1)] + w_{3i-2N_R-2}(k,\ell) \qquad \text{for } i = N_R+1,\ldots,N \end{aligned} \tag{9c}$$

$$\begin{aligned} \mathrm{Im}[e'_i(k,\ell)] = {} & \mathrm{Im}[a^i_{21}]d'_{i-1}(k-1,\ell) + \mathrm{Im}[a^i_{12}]d'_{i-1}(k,\ell-1) + \mathrm{Im}[a^i_{22}]d'_{i-1}(k-1,\ell-1) + \\ & - \mathrm{Im}[b^i_{21}]\mathrm{Re}[e'_i(k-1,\ell)] - \mathrm{Re}[b^i_{21}]\mathrm{Im}[e'_i(k-1,\ell)] - \mathrm{Im}[b^i_{12}]\mathrm{Re}[e'_i(k,\ell-1)] + \\ & - \mathrm{Re}[b^i_{12}]\mathrm{Im}[e'_i(k,\ell-1)] - \mathrm{Im}[b^i_{22}]\mathrm{Re}[e'_i(k-1,\ell-1)] + \\ & - \mathrm{Re}[b^i_{22}]\mathrm{Im}[e'_i(k-1,\ell-1)] + w_{3i-2N_R-1}(k,\ell) \qquad \text{for } i = N_R+1,\ldots,N \end{aligned} \tag{9d}$$

$$\begin{aligned} d'_i(k,\ell) = {} & \mathrm{Re}[e'_i(k,\ell)] + \mathrm{Im}[a^i_{11}]\mathrm{Im}[e'_i(k,\ell)] + \mathrm{Re}[a^i_{21}]\mathrm{Re}[e'_i(k-1,\ell)] + \\ & + \mathrm{Im}[a^i_{21}]\mathrm{Im}[e'_i(k-1,\ell)] + \mathrm{Re}[a^i_{12}]\mathrm{Re}[e'_i(k,\ell-1)] + \mathrm{Im}[a^i_{12}]\mathrm{Im}[e'_i(k,\ell-1)] + \\ & + \mathrm{Re}[a^i_{22}]\mathrm{Re}[e'_i(k-1,\ell-1) + \mathrm{Im}[a^i_{22}]\mathrm{Im}[e'_i(k-1,\ell-1)] - \mathrm{Re}[b^i_{21}]d'_i(k-1,\ell) + \\ & - \mathrm{Re}[b^i_{12}]d'_i(k,\ell-1) - \mathrm{Re}[b^i_{22}]d'_i(k-1,\ell-1) + w_{3i-2N_R}(k,\ell) \qquad \text{for } i = N_R+1,\ldots,N \end{aligned} \tag{9e}$$

$$g'(k,\ell) = d'_N(k,\ell) \tag{9f}$$

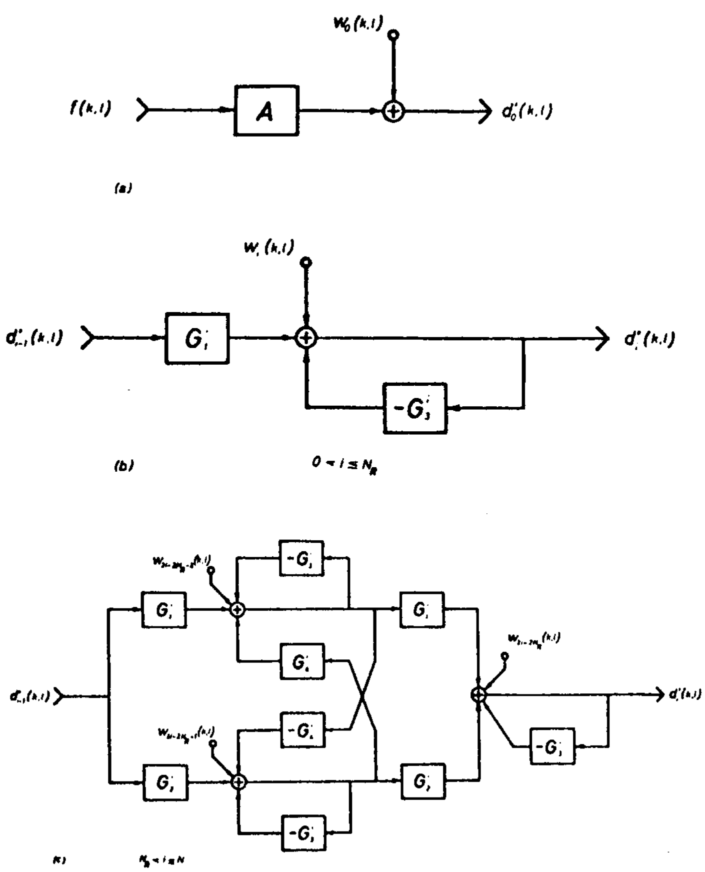

Figure 3 (a)-(c): *Block diagrams for the elements in Figure 1 including the computational errors.*

Taking the two-dimensional z-transform of equations (8) and (9) the block diagrams in Fig. 3 were obtained and the relationship between the computational error $\varepsilon_c(k,\ell)$ and the error sequences $\{w_i(k,\ell)\colon i = 0,1,2,\ldots,3N-2N_R\}$ was determined. This relation is shown in (11) and is interpreted in Fig. 4.

$$E_c(z_1,z_2) = \sum_{i=0}^{3N-2N_R} C_i(z_1,z_2)W_i(z_1,z_2) \tag{11}$$

where the C_i's and G_i's are specified in Table 3.

Equation (11) gives the computational error to be added to the output of the ideal realization to account for the finite wordlength of the actual filter.

In the time domain the total computational error can be equivalently expressed as a sum of double convolutions of the impulse responses of the filters of Fig. 2 and the input noise processes.

We may now assume that the system has reached steady state (stationary noise sources), that the additive noise sources are mutually uncorrelated with zero mean, a flat power spectrum (white noise) and a variance σ_w^2, that is

$$\begin{aligned} \langle w_i(k,\ell)\rangle &= 0 \\ \langle w_i^2(k,\ell)\rangle &= \sigma_w^2 \end{aligned} \qquad \text{for } i = 0,1,2,\ldots,3N-2N_R \tag{12}$$

Here the symbol $\langle\cdot\rangle$ is used to denote statistical average. The total computational error $\varepsilon_c(k,\ell)$ will then be a stationary process with zero mean

$$\langle\varepsilon_c(k,\ell)\rangle \,\hat{=}\, 0.$$

The variance σ_ε^2 can be obtained by application of Parseval's theorem in two dimensions, which in the case where the filters $C_i(z_1,z_2)$ converge on the unit circles in z_1 and z_2, results in

$$\sigma_\varepsilon^2 \triangleq \langle\varepsilon_c^2(k,\ell)\rangle = \frac{\sigma_w^2}{4\pi^2}\sum_{i=0}^{3N-2N_R}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}|C_i(e^{j\mu},e^{j\nu})|^2 d\mu\, d\nu \tag{13}$$

In the general case where the noise sources $\langle w_i(k,\ell)\rangle$ are not white, but stationary with zero mean and power spectral densities $\Phi_{w_i}(z_1,z_2)$ it is straightforward to extend (13). Notice that for a given type of arithmetic, (i.e. fixed-point or floating point, truncation or rounding), the means and variances of the noise sources $\langle w_i(k,\ell)\rangle$ can be determined. Their values can then be used to obtain the mean and variance of the total computational error at the filter output.

In the case of fixed point arithmetic the order in which the second order sections are cascaded can have a great effect on the output computational error, because noise generated in a particular second-order section is filtered by all the succeeding sections. Thus there arises an interesting problem of determining the best pairing of numerator polynomials with denominator polynomials and the test ordering of the filter sections, so that the output signal-to-noise ratio is maximized. The problem is further complicated by the fact that signals must be scaled so that overflow does not occur at any point in the cascade of second-order systems. Jackson has studied problems of that nature but for one-dimensional digital filters only.[20]

TABLE 3

$$C_0(z_1,z_2) = \frac{H(z_1,z_2)}{A}$$

$$C_i(z_1,z_2) = \frac{1}{1+G_3^i}\left[\prod_{j=i+1}^{N_R} H_j(z_1,z_2)\right]\left[\prod_{j=N_R+1}^{N} H_j(z_1,z_2)\,H_j^*(z_1^*,z_2^*)\right] \qquad \text{for } 0 < i \leq N_R$$

$$C_{3i-2N_R-2}(z_1,z_2) = \frac{-G_4^i[G_1^i(1+G_3^i) - G_3^iG_4^i]}{[1+G_3^i]\,[(1+G_3^i)^2-(G_4^i)^2]}\left[\prod_{j=i+1}^{N} H_j(z_1,z_2)\,H_j^*(z_1^*,z_2^*)\right] \qquad \text{for } N_R < i \leq N$$

$$C_{3i-2N_R-1}(z_1,z_2) = \frac{G_2^i(1+G_3^i) + G_1^iG_4^i}{[1+G_3^i]\,[(1+G_3^i)^2+(G_4^i)^2]}\left[\prod_{j=i+1}^{N} H_j(z_1,z_2)\,H_j^*(z_1^*,z_2^*)\right] \qquad \text{for } N_R < i \leq N$$

$$C_{3i-2N_R}(z_1,z_2) = \frac{1}{1+G_3^i}\left[\prod_{j=i+1}^{N} H_j(z_1,z_2)\,H_j^*(z_1^*,z_2^*)\right] \qquad \text{for } N_R < i \leq N$$

$$G_1^i \equiv G_1^i(z_1,z_2) \triangleq 1 + \mathrm{Re}[a_{21}^i]z_1 + \mathrm{Re}[a_{12}^i]z_2 + \mathrm{Re}[a_{22}^i]z_1z_2 \qquad \text{for } i = 1,2,\ldots,N$$

$$G_2^i \equiv G_2^i(z_1,z_2) \triangleq \mathrm{Im}[a_{21}^i]z_1 + \mathrm{Im}[a_{12}^i]z_2 + \mathrm{Im}[a_{22}^i]z_1z_2 \qquad \text{for } i = 1,2,\ldots,N$$

$$G_3^i \equiv G_3^i(z_1,z_2) \triangleq \mathrm{Re}[b_{21}^i]z_1 + \mathrm{Re}[b_{12}^i]z_2 + \mathrm{Re}[b_{22}^i]z_1z_2 \qquad \text{for } i = 1,2,\ldots,N$$

$$G_4^i \equiv G_4^i(z_1,z_2) \triangleq \mathrm{Im}[b_{21}^i]z_1 + \mathrm{Im}[b_{12}^i]z_2 + \mathrm{Im}[b_{22}^i]z_1z_2 \qquad \text{for } i = 1,2,\ldots,N$$

The product term in these equations is set to unity when $\min(j) \geq N$

Realization of two-dimensional IIR filters in a general purpose digital computer

The nature of recursive realizations makes them very suitable for sequentially stored data. Causal filters* are readily implemented. Noncausal filters can be synthesized by means of linear spectral transformations.[13,14] Computer program listings for the design and realization of two-dimensional recursive filters of the type discussed in this paper are given in Ref. 16. Notice that some of the data transformations of Fig. 1 (rotation by 90° and 270° and transpositions) must have access to both rows and columns at the same time for interchange operations, not easily done when the data are stored sequentially in auxiliary storage. The same problem exists with the eight different recursive algorithms of Fig. 1 because each one processes the data in a different order. Therefore, when the amount of memory available is not sufficient to hold the entire array, auxiliary storage is needed and neither of these two techniques is suitable by itself. We have proposed an alternative[15] which results in at most two passes of the data through main memory.

Suppose that we have to implement a filter whose transfer function is a cascade of elementary transfer functions as in (5) and Fig. 2. Assume that the data matrix is too large to be contained in computer main memory and the data are stored sequentially by rows in auxiliary storage, divided into logical records of K rows each. K is chosen so that each submatrix consisting of K consecutive rows can be stored in main memory.

The key point in this implementation is the ordering of the blocks in Fig. 2 corresponding to the cascade of elementary filters in (5). Figure 1 shows all the possibilities for the direction of recursion, sense of recursion and starting point on the input data matrix. Since that matrix has been divided into submatrices of K rows each, which will be processed sequentially, the algorithms whose diagrams have arrows originating in the first row and those with arrows originating in the last row cannot be implemented simultaneously. Therefore, in the block diagram of Fig. 2 we lump together all the filter algorithms whose diagrams have arrows originating in the bottom row (namely, Figs. 2(a), (b), (e) and (h)) followed by a second set of filters whose diagrams have arrows originating in the top row (namely, Figs. 2(c), (d), (f) and (g)). Then successively read each record (submatrix of K rows) into the computer main memory and process it with the first set of filters using the technique of data transformation previously discussed. The result is written back into auxiliary storage. The last output row of each elementary filter must be saved in main memory to serve as the initial conditions for the next record. The same process is done with the remaining records until all the data have been processed. Then the operation is repeated for the second set of filters and with the records in reverse order.

**A two-dimensional causal filter is defined as one that recurses in the (+m, +n) direction (cf. (2)).*

Thus, this technique is compatible with the sequential access of the data and at most two passes of the data through main memory are necessary. It should be noted that while two passes are necessary in implementing filters with zero phase response, only one pass would be required in some cases where the zero phase response requirement is not necessary.

Examples of impulse responses of two-dimensional recursive filters are shown in Fig. 5 through 10. The filter coefficients were obtained as described under "Control of the cutoff frequencies". To calculate the impulse responses equations (8) were solved in each case for 41 × 41 discrete input values, where the only non-zero value was f(21 21) = 1.0 and zero initial conditions were always assumed. Figures 7 through 10 show impulse responses existing in more than one quadrant because they were obtained with non-causal filters. The non-causal filters were realized by combining rotated filters and data rotations (i.e. rotations of the matrix containing the data by multiples of 90°), as shown in Ref. 7. The filters shown in Figs. 9 and 10 have zero-phase response and their magnitude responses are the square of the magnitude responses of the filters shown in Figs. 7 and 8, respectively. An example of a typical magnitude response is shown in Fig. 11 and more examples are shown in Ref. 16.

TABLE 4

CPU time in seconds employed in computing the impulse responses

Figure No.	5	6	7	8	9	10
CPU Time in Seconds	0.13	0.35	0.30	0.55	0.73	1.42
M (see Eq. 5)	2	6	4	12	8	24
Number of Data Rotations	0	0	2	2	4	4

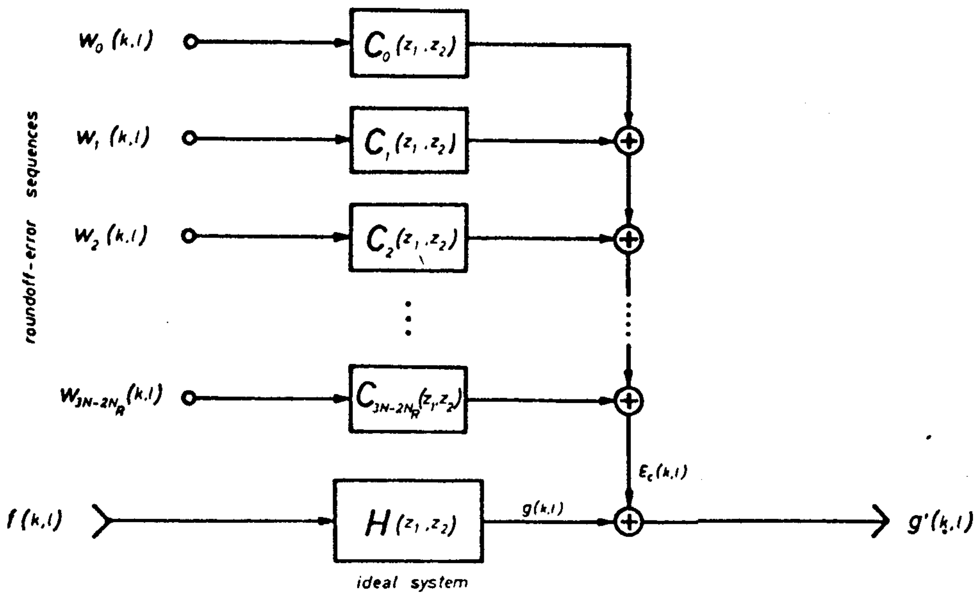

Figure 4: *Noise model in two-dimensional complex cascade programming.*

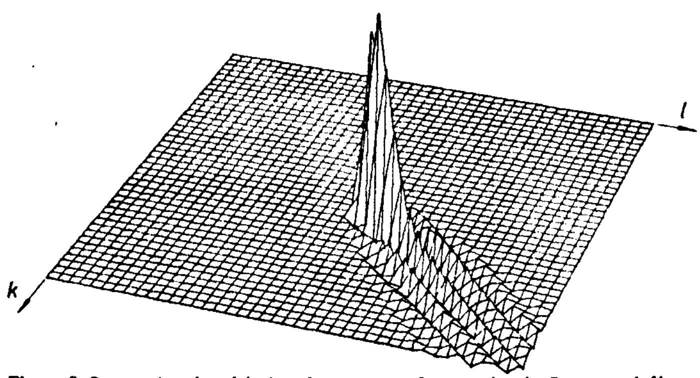

Figure 5: *Perspective plot of the impulse response of a second-order Butterworth filter rotated 315°.*

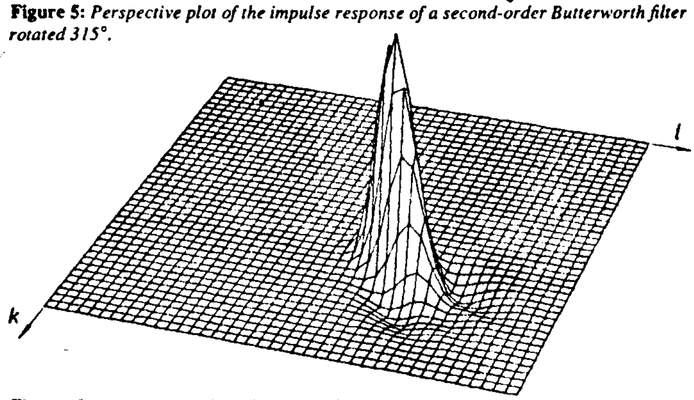

Figure 6: *Perspective plot of the impulse response of an elliptically-shaped filter formed by cascading three second-order Butterworth filters rotated 285°, 315° and 345°.*

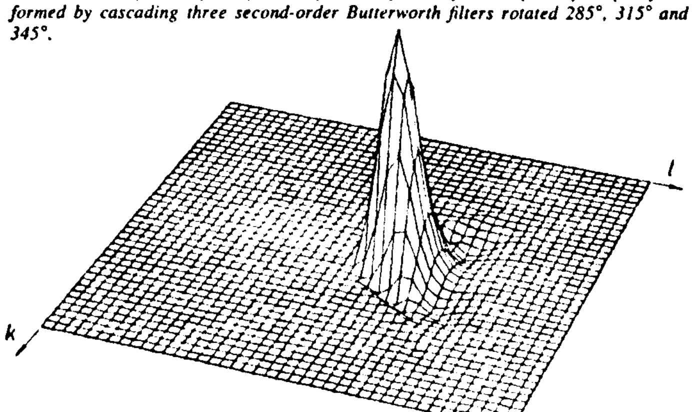

Figure 7: *Perspective plot of the impulse response of a cascade of two second-order Butterworth filters rotated 225° and 315°.*

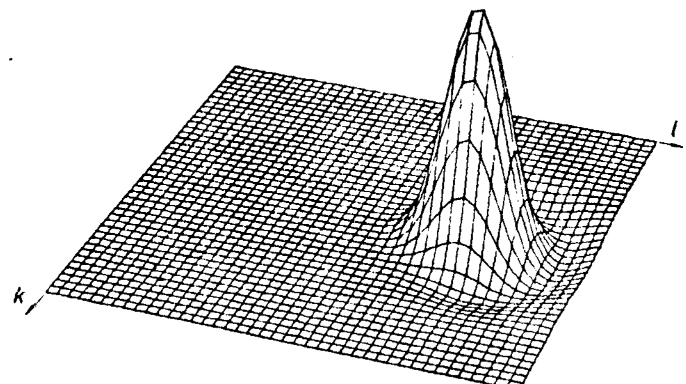

Figure 8: *Perspective plot of the impulse response of a cascade of six second-order Butterworth filters rotated by multiples of 30°.*

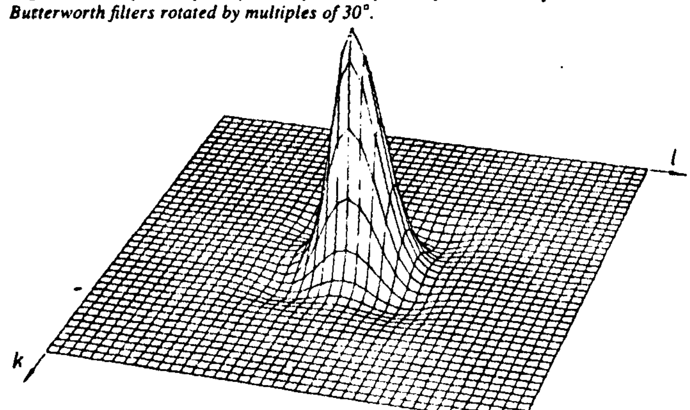

Figure 9: *Perspective plot of the impulse response of a filter with zero-phase response formed by cascading four second-order Butterworth filters rotated by multiples of 90°.*

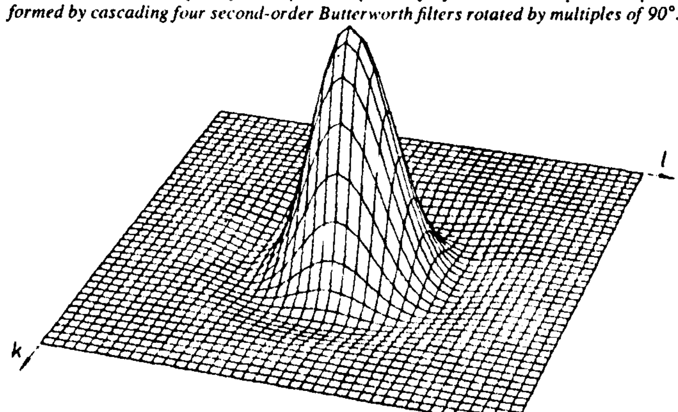

Figure 10: *Perspective plot of the impulse response of a filter with zero-phase response formed by cascading twelve second-order Butterworth filters rotated by multiples of 30°.*

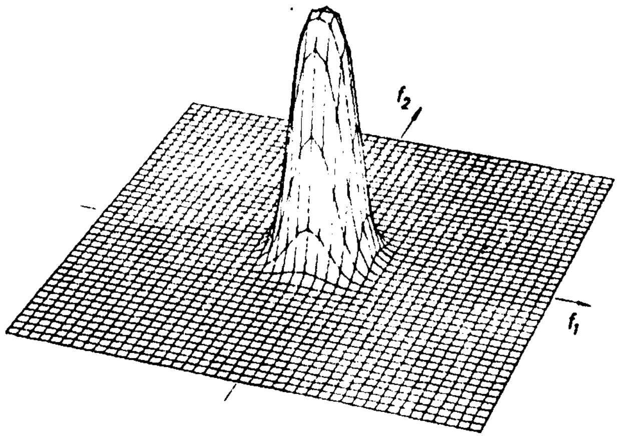

Figure 11: *Magnitude response of the filter whose impulse response is shown in Figure 10.*

Equations (8) were coded in FORTRAN IV. The source listings are given in Ref. 16. The machine used for the computations was the IBM SYSTEM/370-165 with a FORTRAN IV (H Extended) compiler and the perspective plots were produced with a CALCOMP plotter. Table 4 gives the CPU time employed in obtaining the data for Figs. 5 through 10. The CPU times given here include the time necessary for data rotations.

Conclusions

A summary of design techniques for two dimensional recursive filters was presented.

The problem of controlling the cutoff frequency was discussed and a new algorithm useful for achieving given specifications was proposed. The technique of complex cascade programming was adapted for the implementation of factorable two-dimensional IIR filters. Although the number of multiplies for complex cascade programming is higher than those required if each pair of complex sections were to be replaced by a fourth order section with real coefficients, this method has still some advantages.

The computational errors in two-dimensional complex cascade programming were discussed. It was shown that these errors can be represented by a set of error sequences additively contaminating the output of an ideal realization. The transfer functions relating these error sequences to simple roundoff errors were derived and an expression was obtained for the error that should be added to the filter output of the ideal realization to account for the finite wordlength of the actual filter. Finally, the problems of data management when large matrices are stored in auxiliary storage were briefly considered. These techniques have been used in the design of digital filters for the tomographic enhancement of radiographs.[15]

References

1. R. M. Mersereau and D. E. Dudgeon, "Two-Dimensional Digital Filtering", *Proc. of the IEEE*, Vol. 63, No. 4, pp. 610-623, April 1975.
2. S. Chakrabarti and S. K. Mitra, "Design of Two-Dimensional Digital Filters Via Spectral Transformations", *Proc. of the IEEE*, Vol. 65, No. 6, pp. 905-914, June 1977.
3. E. L. Hall, "A Comparison of Computations for Spatial Frequency Filtering", *Proc. of the IEEE*, Vol. 60, No. 7, pp. 887-891, July 1972.
4. T. S. Huang, W. F. Schreiber and O. J. Tretiak, "Image Processing", *Proc. of the IEEE*, Vol. 59, No. 11, pp. 1586-1609, November 1971.
5. J. L. Shanks and R. R. Read, "Two-Dimensional Recursive Filters for Digital Processing", *Proc. of the 1973 International Symposium on Circuit Theory*, Toronto, Ontario, Canada, pp. 32-35, April 1973.
6. S. K. Mitra, A. D. Sagar and N. A. Pendergrass, "Realizations of Two-Dimensional Recursive Digital Filters", *IEEE Trans. on Circuits and Systems*, Vol. CAS-22, No. 3, pp. 177-184, March 1975.
7. J. M. Costa and A. N. Venetsanopoulos, "Design of Circularly Symmetric Two-Dimensional Recursive Filters", *IEEE Trans. on Acoustics, Speech and Signal Processing*, Vol. ASSP-22, No. 6, pp. 432-443, December 1974.
8. C. J. Harrison, "The Design and Application of Two-Dimensional Digital Filters", M.Sc. Thesis, Department of Geophysics, University of Western Ontario, London, Ontario, May 1978.
9. T. S. Huang, "Stability of Two-Dimensional Recursive Filters", *IEEE Trans. on Audio and Electroacoustics*, Vol. AU-20, No. 2, pp. 158-163, June 1972.
10. J. L. Shanks, "Two-Dimensional Recursive Filters", in *1969 SWIEEECO Record*, pp. 19E1-19E8, April 1969.
11. J. L. Shanks, S. Treitel, and J. H. Justice, "Stability and Synthesis of Two-Dimensional Recursive Filters", *IEEE Trans. on Audio and Electroacoustics*, Vol. AU-20, No. 2, pp. 158-163, June 1972.
12. D. J. Goodman, "A Design Technique for Circularly Symmetric Low-Pass Filters", *IEEE Trans. on Acoustics, Speech and Signal Processing*, Vol. ASSP-26, No. 4, pp. 290-304, August 1978.
13. J. M. Costa and A. N. Venetsanopoulos, "A Group of Linear Spectral Transformations for Two-Dimensional Digital Filters", *IEEE Trans. on Acoustics, Speech and Signal Processing*, Vol. ASSP-24, No. 5, pp. 424-425, October 1976.
14. A. N. Venetsanopoulos and J. M. Costa, "Design of Two-Dimensional Digital Filters", *Proceedings of the First International Conference on Information Sciences and Systems*, University of Patras, Greece, Vol. 3, pp. 820-825, August 19-24, 1976.
15. J. M. Costa, "Design and Realization of Digital Tomographic Filters for Radiographs", Ph.D. Dissertation, Department of Electrical Engineering, University of Toronto, Toronto, Ontario, Canada, (in preparation).
16. J. M. Costa and A. N. Venetsanopoulos, "Recursive Implementation of Factorable Two-Dimensional Filters", Communications Technical Report No. 78-1, Department of Electrical Engineering, University of Toronto, Toronto, Ontario, Canada, March 1978, p. 20.
17. J. B. Knowles and R. Edwards, "Complex Cascade Programming and Associated Computational Errors", *Electronic Letters*, Vol. 1, No. 6, pp. 160-161, August 1965.
18. G. A. Maria and M. M. Fahmy, "An ℓ_p Design Technique for Two-Dimensional Digital Recursive Filters", *IEEE Trans. on Acoustics, Speech and Signal Processing*, Vol. ASSP-22, No. 1, pp. 15-21, February 1974.
19. J. B. Knowles and R. Edwards, "Effect of Finite-Wordlength Computer in a Sampled-Data Feedback System", *Proc. of the IEE*, Vol. 112, No. 6, pp. 1197-1207, June 1965.
20. L. B. Jackson, "On the Interaction of Roundoff Noise and Dynamic Range in Digital Filters", *Bell System Telephone Journal*, Vol. 49, pp. 159-184, February 1970.