

DESIGN AND REALIZATION OF STABLE TWO-DIMENSIONAL RECURSIVE FILTERS

by

José M. COSTA Vela

A thesis submitted in conformity with
the requirements for the degree of
Master of Applied Science
in the University of Toronto

Department of Electrical Engineering

September, 1973

ABSTRACT

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. A review of the theory of two-dimensional digital filters is presented. This includes a brief discussion of some of the computational algorithms used and the state of the art in the design of two-dimensional digital filters. Some basic kinds of filters required for image processing are described together with their applications. A technique is presented for designing stable two-dimensional recursive filters whose magnitude response is approximately circularly symmetric. This is achieved by cascading a number of elementary filters which are called rotated filters because they are designed by rotating one-dimensional continuous filters and using the two-dimensional z-transform to obtain the corresponding digital filter. Stability of these filters is considered in detail and the results obtained are stated in two corollaries. In particular it is proved that rotated filters are stable if the angle of rotation is between 270° and 360° . For the realization of those filters, the technique of complex cascade programming is developed in two dimensions. Examples are given. The computational errors in two-dimensional complex cascade programming are analyzed. Finally, methods of analysis and design of the shape, circular symmetry and cutoff frequency of two-dimensional recursive filters are shown.

ACKNOWLEDGEMENT

to Professor A.N. Venetsanopoulos
for his assistance and
encouragement throughout the
course of this work.

TABLE OF CONTENTS

ABSTRACT	ii
ACKNOWLEDGMENT	iii
TABLE OF CONTENTS	iv
LIST OF PRINCIPAL SYMBOLS	vi
I. INTRODUCTION	1
II. TWO-DIMENSIONAL DIGITAL FILTERS	2
III. APPLICATIONS OF TWO-DIMENSIONAL DIGITAL FILTERS IN IMAGE PROCESSING	9
IV. DESIGN OF STABLE DIGITAL FILTERS	12
1. Design of Rotated Filters	12
2. Stability of Rotated Filters	16
3. Design of Filters with Circular Symmetry	23
V. REALIZATION	26
1. Two-Dimensional Complex Cascade Programming	26
2. Computational Errors in Two-Dimensional Complex Cascade Programming	30
VI. CHOICE OF FILTER PARAMETERS	34
1. Cutoff frequency	34
2. Shape factors	39
VII. CONCLUSIONS	41
APPENDIX A	A.1-A.6
APPENDIX B	B.1-B.2
APPENDIX C	C.1-C.5

APPENDIX D	D.1-D.2
APPENDIX E	E.1-E.2
APPENDIX G	G.1-G.29
REFERENCES	R.1-R.3
FIGURES 1 to 32, A.1, and C.1	F.1-F.40
TABLES I to XVIII	T.1-T.22

LIST OF PRINCIPAL SYMBOLS

a_{ij}	coefficients of the numerator of the transfer function of a two-dimensional digital filter	4,6
$A(z_1, z_2)$	numerator of the transfer function of a two-dimensional digital filter	6
b_{ij}	coefficients of the denominator of the transfer function of a two-dimensional digital filter	4,6
$B(z_1, z_2)$	denominator of the transfer function of a two-dimensional digital filter	6
c	centre of the circle image of the unit circle by a bilinear transformation	17
$C_i(z_1, z_2)$	transfer functions for computational errors in complex cascade programming	33
$d_i(k, \ell)$	ideal output sequences from the filter sections realized using complex cascade programming	28
$d'_i(k, \ell)$	actual output sequences from the filter sections realized using complex cascade programming	31
$e_i(k, \ell)$	ideal intermediate sequences in the filter sections realized using complex cascade programming	28
$e'_i(k, \ell)$	actual intermediate sequences in the filter sections realized using complex cascade programming	31
f_1	frequency in the x direction	9
f_2	frequency in the y direction	9
f_c	cutoff frequency of a filter given as a fraction of the Nyquist frequency	37
f_n	Nyquist frequency	16
f_u	desired cutoff frequency of a filter given as a fraction of the Nyquist frequency	37
f_{r1}	frequency in the x direction given as a fraction of the Nyquist frequency	16

f_{r_2}	frequency in the y direction given as a fraction of the Nyquist frequency	16
$f(z_1)$	bilinear transformation	17
$f(m,n)$	discrete two-dimensional signal or two-dimensional sequence	2
$f(m,n)$	input to a two-dimensional filter	2
$f(x,y)$	continuous two-dimensional signal	2
$F(u,v)$	two-dimensional Fourier transform of the sequence $f(m,n)$	4
$F(z_1, z_2)$	two-dimensional z-transform of the sequence $f(m,n)$	9
$g(m,n)$	output of a two-dimensional digital filter	2
G_j^i	transfer functions of the elements in the filter sections using two-dimensional complex cascade programming	34
$h(m,n)$	impulse response of a two-dimensional digital filter	2
$H(s)$	transfer function of a one-dimensional continuous filter	12
$H(v)$	transfer function of a circularly symmetric two-dimensional digital filter	10
$H(s_1, s_2)$	transfer function of a two-dimensional continuous filter	13
$H(z_1, z_2)$	transfer function of a two-dimensional digital filter	6
Im	imaginary part of a complex quantity	28
p_i	pole locations of a one-dimensional filter	12
q_i	zero locations of a one-dimensional filter	12
r	radius of the circle image of the unit circle by a bilinear transformation	17
Re	real part of a complex quantity	28
T	sampling period (the same in the two directions)	9,14
T	linear transformation, usually a rotation by 90°	24
$U_0(m,n)$	unit impulse function	3
$w_i(k,\ell)$	roundoff errors	31

$w_i(z_1, z_2)$	two-dimensional z-transform of $w_i(k, \ell)$	33
X	sampling period in the x direction	2
Y	sampling period in the y direction	2
z_1	unit delay operator in the x direction	5
z_2	unit delay operator in the y direction	5
α	parameter of the two-dimensional frequency transformation	35
β	angle of rotation of a filter or a matrix of data	13
β_d	angle of rotation of an input data matrix (a multiple of 90°)	22
β_e	effective angle of rotation of a filter with respect to the original input data matrix	22
β_f	angle of rotation of a filter	22
$\epsilon_c(k, \ell)$	computational error on the computer output using complex cascade programming	32
$\epsilon_i(k, \ell)$	computational error quantities in two-dimensional complex cascade programming	32
$\Sigma_c(z_1, z_2)$	two-dimensional z-transform of $\epsilon_c(k, \ell)$	33
ω_1	angular frequency in the x direction	6
ω_2	angular frequency in the y direction	6
ω_c	cutoff angular frequency of a filter	35
ω_u	desired cutoff angular frequency in the design of a filter	35
v	longitudinal polar-coordinate	10
θ	angular polar-coordinate	10
*	complex conjugate	16
u	or 20	
n	and 20	

I. INTRODUCTION

There are many signals inherently two-dimensional in nature, such as photographic data (weather photos, air reconnaissance photos, medical radiographs, and nuclear medicine images) and geophysical data (seismic records, potential field survey records, gravity and magnetic data).

These signals can be processed by coherent optical systems or by digital computer. A comparison between the advantages and limitations of both techniques is given in [1]. This comparison was done on the basis of flexibility, capacity, speed, accuracy, and cost. The conclusions are that the main advantages of a coherent optical system are its information storage capacity and processing speed, and the main advantages of a digital computer are its flexibility and accuracy. Coherent optical systems are suitable for doing linear operations, such as Fourier transformation and linear filtering, on large-volume data; but when nonlinear operations or accurate linear operations on a limited amount of data are required, digital computers can be used to advantage. In some cases, although the filtering is best done by a coherent optical system, the spatial filter is most conveniently made on a digital computer. In this **thesis** we consider only the digital processing of two-dimensional signals by computer.

Most existing techniques for processing one-dimensional signals can be extended to two dimensions. However, in some cases important problems of stability and synthesis arise. In Section II we review the theory of two-dimensional filters, discussing some computational algorithms and commenting on the state of the art in the design of two-dimensional digital filters.

Applications to the processing of images are considered in Section III, describing some of the more important operations which are desirable to perform on images leading to the use of two-dimensional filters.

In Section IV we present a procedure for designing stable recursive filters whose frequency response is approximately circularly symmetric.

The recursive realization of these filters and its associated computational errors are developed in Section V. Examples are given.

Finally, Section VI contains methods of analysis and design of the shape, circular symmetry and cutoff frequency for the filters in Section IV.

II. TWO-DIMENSIONAL DIGITAL FILTERS

A two-dimensional signal is a function $f(x,y)$ of two variables, usually time and/or space. In the case of a picture both variables would be spatial, in the case of seismic records the variable in one direction would be space, and in the other direction, time.

Sampling such a signal, $f(x,y)$, at equi-spaced intervals, X and Y , in the x and y directions, respectively, a discrete signal or sequence $f(m,n)$ results

$$f(m,n) \stackrel{\Delta}{=} f(mX, nY) = f(x,y) \Big|_{x=mX, y=nY} \quad (1)$$

where m and n are integer variables. Assume that the sampling rates, $1/X$ and $1/Y$, are fast enough to prevent aliasing.

A linear position-invariant two-dimensional digital filter may be defined by the (double) convolution relation as

$$g(m,n) = h(m,n) * f(m,n) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} h(i,j) f(m-i, n-j) \quad (2)$$

where $f(m,n)$ and $g(m,n)$ are the input and output, respectively, of the

filter. The weights $h(m,n)$ define the two-dimensional filter. Also, if the input is the unit impulse, defined by

$$U_0(m,n) = \begin{cases} 1 & \text{for } m = n = 0 \\ 0 & \text{otherwise} \end{cases}$$

then, the output is $h(m,n)$ and is referred to as the impulse response or point-spread function of the filter.

A filter is stable if every bounded (finite) input produces a bounded (i.e., finite) output. Theorem 1 in Appendix A states and proves that a necessary and sufficient condition for stability is that

$$\sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} |h(m,n)| < \infty \quad (3)$$

Note that this equation may be quite difficult to evaluate for arbitrary $h(m,n)$. Alternate conditions for stability are also given in Appendix A.

A two-dimensional filter is said to be separable if its impulse response can be factored into a product of one-dimensional responses, that is,

$$h(m,n) = h_1(m) h_2(n) \quad (4)$$

The advantage of separable filters is that the two-dimensional convolution (2) can be carried out as a sequence of one-dimensional convolutions.

When both the input sequence and the impulse response of the filter are nonzero only over a finite area, say from $0 \leq m \leq M - 1$ and $0 \leq n \leq N - 1$, then the filtering operation may be performed via the

FFT, by computing the transform of the input signal, multiplying by the transform of the filter impulse response, and inverse transforming the result. In two-dimensions the discrete Fourier transform (DFT) of a sequence $f(m,n)$ is given by

$$F(u,v) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) \exp [-j2\pi (\frac{m}{M} u + \frac{n}{N} v)] \quad (5)$$

with

$$\begin{aligned} u &= 0, 1, \dots, M-1 \\ v &= 0, 1, \dots, N-1 \end{aligned} \quad (6)$$

A digital filtering operation can also be described by a linear difference equation. The general form is

$$g(m,n) = \sum_{i=1}^{M_a} \sum_{j=1}^{N_a} a_{ij} f(m-i+1, n-j+1) - \sum_{i=1}^{M_b} \sum_{j=1}^{N_b} b_{ij} g(m-i+1, n-j+1) \quad i, j \neq 1 \text{ simultaneously} \quad (7)$$

Here we assume either that all output values $g(m-j+1, n-j+1)$ have been computed previously or are equal to zero (initial conditions).

A filter realized via recursive relations of the form of (7) is called a recursive filter; since the output is an explicit function of past output samples as well as past and/or present input samples. In a nonrecursive filter the output samples of the filter are explicitly determined as a weighted sum of past and present input samples only.

Equation (7) can be rewritten in the form

$$\sum_{i=1}^{M_b} \sum_{j=1}^{N_b} b_{ij} g(m-i+1, n-j+1) = \sum_{i=1}^{M_a} \sum_{j=1}^{N_a} a_{ij} f(m-i+1, n-j+1) \quad (8)$$

where $b_{11} = 1$.

Equation (8) can be solved for $g(m,n)$ (7), $g(m-M_b+1, n)$, $g(m, n-N_b+1)$, 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.

A recursive filter is said to be causal if it recurses in the $(+m, +n)$ direction [15]. A causal filter is realizable if its impulse response satisfies the property

$$h(m,n) = 0 \quad m,n < 0$$

A useful representation of (8) is obtained by the two-dimensional z-transform or double z-transform. This transform is defined as follows

$$X(z_1, z_2) \triangleq \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x(m,n) z_1^m z_2^n \quad (9)$$

where $x(m,n)$ is a two-dimensional sequence and $X(z_1, z_2)$ its two-dimensional z-transform (*). The domain 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 the properties of the z-transform in one dimension.

(*) Some authors use an equivalent definition by replacing the unit delay operators z_1 and z_2 in (9) by z_1^{-1} and z_2^{-1} , respectively. We use the definition in (9) because it is most common in the two-dimensional filter literature [3], [15].

Depending upon whether $h(m,n)$ has an infinite number of terms or not, digital filters are classified as infinite impulse response filters (IIR) or finite impulse response filters (FIR). For FIR filters (3) is always satisfied for finite $|h(m,n)|$, but for IIR filters it may be difficult to determine whether or not a given filter is stable.

The implementation of FIR filters is usually done by direct convolution (2) or by FFT techniques and IIR filters may be realized in direct form by implementing a recursive difference equation such as (7) or more effectively combining in cascade or in parallel form lower order filters.

A comparison of the computational algorithms for two-dimensional digital filters in terms of speed and memory usage is developed in [2].

In the remaining of this section we summarize the state of the art in the design of two-dimensional digital filters.

The problem of designing FIR two-dimensional digital filters in the space-time domain can be stated as follows: Given the desired impulse response in the form of a two-dimensional array or matrix of the weighting coefficients find a filter which approximates that impulse response. A nonrecursive filter could be readily implemented using straight convolution or the FFT, but if the dimensions of the matrix representing the impulse response are large this realization becomes computationally inefficient. Shanks [3] has extended a time-domain synthesis technique of recursive filters in two-dimensions. The resulting filter is nonseparable. Another approach (developed in references [4]-[7]) consists of expanding the given coefficient matrix into a finite and converging sum of matrices having the same dimensions

as the original filter impulse response array, such that each constituent matrix of this sum is separable, in the sense that it can be expressed as the product of a column vector multiplied with a row vector (cf.(4)). Each term, or filter stage, can then be approximated by an appropriate one-dimensional filter, recursive or nonrecursive. In many cases the given matrix of coefficients can be represented with small error with substantial gains in computing efficiency by keeping only the first few terms of this expansion.

In the frequency domain the design techniques for FIR filters in one-dimension can be often extended to two-dimensions because these filters are guaranteed to be stable. Huang [8] has shown that the windowing technique may be extended to two dimensions by forming two-dimensional circularly symmetric windows from one-dimensional windows and Hu [9] has developed a design technique for filters with circularly symmetric frequency response by extending the techniques known for frequency sampling and optimal (equiripple) FIR filters.

On the other hand, the design of IIR filters in two dimensions becomes very difficult, mainly for two reasons:

- 1) A polynomial in two variables, $P(z_1, z_2)$, cannot in general be factored into first and 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 realized in parallel or cascade form to reduce the effect of quantization noise. In Section IV we develop a design procedure for two-dimensional recursive filters which will factor into second-order filters.

2) It is very difficult to test the stability of two-dimensional IIR filters except for simple filters. In Appendix A we review the most relevant theorems concerning stability of two-dimensional filters. The filters designed in Section IV are guaranteed to be stable, their stability is considered in Section IV.2.

Farmer and Gooden [10] and Shanks [3] have presented examples of a technique showing how to convert one-dimensional filters into two-dimensional filters with arbitrary directivity in a two-dimensional frequency response plane. Corollary 2 in Section IV.2 solves the problem of determining the stability of those filters beforehand.

More recent papers, not directly related to this thesis, can be found in [20]-[23].

III. APPLICATIONS OF TWO-DIMENSIONAL DIGITAL FILTERS IN IMAGE PROCESSING

In image processing there is no preferred spatial frequency axis. It is therefore desirable to process images with filters whose frequency response approximates a circularly symmetric function satisfying (12)

$$H\left(e^{-j2\pi f_1 T}, e^{-j2\pi f_2 T}\right) \approx H\left(e^{-j2\pi (f_1^2 + f_2^2)^{1/2} T}\right) \quad (12)$$

where it is assumed that the sample interval T is the same in both directions.

Nevertheless, using a rectangular sampling grid, the requirement of circular symmetry cannot be exactly met. However, very good approximations to (12) can be obtained in practice.

For a frequency response circularly symmetric around the origin it is convenient to adopt a polar-coordinate system (v , θ), where $v = \sqrt{f_1^2 + f_2^2}$ is measured as cycles per unit length.

The general objective of image enhancement is to make selected features easier to see. This might require suppression of useless data such as random noise and background shading or perhaps amplification of fine detail. The types of filters most used in image enhancement are low-pass, high-pass, and high-emphasis filters [12]. Applications of other types of digital filters to picture processing may be found in [1], [13], and [14].

The magnitude response $|H(v)|$ of a low-pass filter is similar to that of a one-dimensional filter, and the magnitude responses of a high-pass $|H'(v)|$ and a high-emphasis $|H''(v)|$ filters can be derived from that of a low-pass filter as follows:

$$|H'(v)| = 1 - |H(v)| \quad (13)$$

$$|H''(v)| = 1 + A|H'(v)| \quad (14)$$

where A is the gain at high frequencies.

A low-pass filter passes the low-frequency signal components and rejects the high ones. Removal of high frequency components from an image may be desirable in several situations. The most common application is made to pictures containing excessive random noise which makes large low-contrast features difficult to see clearly. Sometimes

it is useful to remove high-frequency structure, such as sharp edges, that are not important and make the rest of the picture difficult to view. For example the random noise in a radiograph results from the spatial fluctuations of the illuminating radiation and film scanning systems which inject noise into the image. This noise is due to fluctuations in the light source and electrical noise in the output of the light-sensing device.

A high-pass filter has the opposite function of a low-pass filter. It removes the low-frequency signals and passes the high-frequency signals. A common application of high-pass filters is in radiographs where it is difficult to visualize low-contrast features when they are superimposed onto a very dark or very light background. For example, the small bones in the ear cannot generally be seen in a standard x-ray film, because they absorb too little radiation relative to the larger surrounding bone mass. The high-pass filter removes the background by converting constant or very slowly changing dark or light areas to grey.

A high-emphasis or high-frequency restoration filter passes low-frequency signals unchanged and amplifies high-frequency signals. Applied to a picture, this type of filter sharpens edges and generally magnifies small detail. A high-pass filter and a high-emphasis filter are similar except the high-pass filter removes the low-frequencies while the high-emphasis does not.

IV. DESIGN OF STABLE RECURSIVE FILTERS

In this section we present a design procedure to obtain two-dimensional recursive filters whose frequency response approximates a circularly symmetric function and that are guaranteed to be stable.

This approach approximates the circular shaped contour levels of the magnitude response by a polygonal shape. This is achieved by cascading a number of elementary filters which have specific directivities in the two-dimensional frequency response plane. These elementary filters are designed according to the technique of Shanks [3]. A generalization of the design procedure and stability considerations follow.

1. Design of Rotated Filters

Suppose a one-dimensional continuous filter whose impulse response is real, is given in its factored form

$$H_1(s) = H_0 \frac{\prod_{i=1}^m (s-q_i)}{\prod_{i=1}^n (s-p_i)} \quad (15)$$

where H_0 is a scalar gain constant. The zero locations q_i and the pole locations p_i may be complex, in which case their conjugates are also present in the corresponding product.

The filter given in (15) can also be viewed as a two-dimensional filter that varies in one dimension only and could be written as follows

$$H_2(s_1, s_2) = H_1(s_2) = H_0 \frac{\prod_{i=1}^m (s_2 - q_i)}{\prod_{i=1}^n (s_2 - p_i)} \quad (16)$$

Rotating clockwise the (s_1, s_2) axes through an angle β by means of the transformation (17) we obtain the filter of (18), whose frequency response is rotated by an angle $(-\beta)$ with respect to the frequency response of (16)

$$s_1 = s'_1 \cos \beta + s'_2 \sin \beta \quad (17a)$$

$$s_2 = -s'_1 \sin \beta + s'_2 \cos \beta \quad (17b)$$

$$H_2(s'_1, s'_2) = H_0 \frac{\prod_{i=1}^m [(s'_2 \cos \beta - s'_1 \sin \beta) - q_i]}{\prod_{i=1}^n [(s'_2 \cos \beta - s'_1 \sin \beta) - p_i]} \quad (18)$$

$H_2(s'_1, s'_2)$ describes a continuous two-dimensional filter in the new coordinate system of s'_1 and s'_2 . To produce the corresponding two-dimensional discrete filter we use the two-dimensional bilinear z-transform [3] defined by the following two equations:

$$s'_1 = \frac{2}{T} \frac{1 - z_1}{1 + z_1} \quad (19a)$$

$$s'_2 = \frac{2}{T} \frac{1 - z_2}{1 + z_2} \quad (19b)$$

It is assumed throughout the study that the sample interval T is the same in both directions. Substituting (19) into (18) we obtain

$$H(z_1, z_2) = A \prod_{i=1}^M \frac{a_{11}^i + a_{21}^i z_1 + a_{12}^i z_2 + a_{22}^i z_1 z_2}{b_{11}^i + b_{21}^i z_1 + b_{12}^i z_2 + b_{22}^i z_1 z_2} \quad (20)$$

where

$$A = H_0 \left(\frac{T}{2}\right)^{n-m}$$

$$M = \max(m, n)$$

$$a_{11}^i = \cos \beta - \sin \beta - \frac{T}{2} q_i$$

$$a_{21}^i = \cos \beta + \sin \beta - \frac{T}{2} q_i$$

$$a_{12}^i = -\cos \beta - \sin \beta - \frac{T}{2} q_i$$

$$a_{22}^i = -\cos \beta + \sin \beta - \frac{T}{2} q_i \quad \text{for } 1 \leq i \leq m$$

$$a_{11}^i = a_{21}^i = a_{12}^i = a_{22}^i = 1 \quad \text{for } m < i \leq M$$

$$b_{11}^i = \cos \beta - \sin \beta - \frac{T}{2} p_i$$

$$b_{21}^i = \cos \beta + \sin \beta - \frac{T}{2} p_i$$

$$b_{12}^i = -\cos \beta - \sin \beta - \frac{T}{2} p_i$$

$$b_{22}^i = -\cos \beta + \sin \beta - \frac{T}{2} p_i \quad \text{for } 1 \leq i \leq n$$

$$b_{11}^i = b_{21}^i = b_{12}^i = b_{22}^i = 1 \quad \text{for } n < i \leq M$$

Equation (20) can be physically interpreted as a cascade of bilinear second-order systems and can be readily realized by complex cascade programming as shown in Section V. 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.

Examples of rotated filters are given in Figs. 1 through 4. These figures show contour maps of the magnitude responses of a

second-order Butterworth filter rotated through different angles.

Frequencies are shown as fractions of the Nyquist frequency, that is,

$$f_{r_1} = \frac{f_1}{f_n} \quad \text{and} \quad f_{r_2} = \frac{f_2}{f_n}$$

where $f_n \stackrel{\Delta}{=} 1/2T$ is the Nyquist frequency. In every case the magnitude response has been normalized to a peak response of 1.0, the contour interval is 0.1, and the cutoff frequency is 0.2. The distortion at high frequencies is caused by the fact that the two-dimensional bilinear z-transform makes the magnitude response to be zero along the Nyquist frequency in each axis. In Fig. 5 we show fine details of the contour plot in Fig. 3 at high frequencies.

2. Stability of Rotated Filters

Some basic theorems concerning stability of two-dimensional recursive filters are reviewed in Appendix A. Using Theorem 3 the conditions that the coefficients in (20) must satisfy to ensure the stability of the filter were derived. The results are given in the following two corollaries.

Corollary 1: A causal recursive filter with transfer function

$$H(z_1, z_2) = \frac{A(z_1, z_2)}{B(z_1, z_2)} \cdot \frac{A^*(z_1^*, z_2^*)}{B^*(z_1^*, z_2^*)} \quad (22)$$

where $A(z_1, z_2)$ and $B(z_1, z_2)$ are complex polynomials in z_1 and z_2 and

$B(z_1, z_2)$ is of the form

$$B(z_1, z_2) = b_{11} + b_{21} z_1 + b_{12} z_2 + b_{22} z_1 z_2$$

is stable if

$$\left| |b_{21} b_{22}^* - b_{11} b_{12}^*| - |b_{11} b_{22} - b_{21} b_{12}| \right| > \left| |b_{12}|^2 - |b_{22}|^2 \right| \quad (23)$$

and

$$|b_{11}| > |b_{21}| \quad (24)$$

Proof: We want to show that for the class of filters specified in this Corollary, (23) and (24) are equivalent to the "if" part of Theorem 3 in Appendix A.

Setting $B(z_1, z_2) = 0$ we obtain

$$z_2 = -\frac{b_{11} + b_{21} z_1}{b_{12} + b_{22} z_1} \stackrel{\Delta}{=} f(z_1) \quad (25)$$

which is a bilinear transformation mapping circles into circles (*)

Clearly, for the filter $\frac{A(z_1, z_2)}{B(z_1, z_2)}$ condition 1) of Theorem 3 is satisfied if and only if

$$\left| |c| - r \right| > 1 \quad (26)$$

where c is the centre and r is the radius of the circle image. We have derived expressions for c and r as a function of the filter coefficients

(*) Here we include straight lines as circles with infinite radii.

in Lemma 1 of Appendix C.

But (26) is equivalent to (23) by substituting (C.2) and (C.3) into (26), that is

$$\left| \frac{\left| b_{21} b_{22}^* - b_{11} b_{12}^* \right|}{\left| b_{12} \right|^2 - \left| b_{22} \right|^2} - \frac{\left| b_{11} b_{22} - b_{21} b_{12} \right|}{\left| b_{12} \right|^2 - \left| b_{22} \right|^2} \right| > 1$$

which is equivalent to (23)

For condition 2) consider the inverse transformation

$$z_1 = f^{-1}(z_2) = - \frac{b_{11} + b_{12} z_2}{b_{21} + b_{22} z_2}$$

$$\hat{z}_1 \triangleq f^{-1}(z_2) \Big|_{z_2=0} = - \frac{b_{11}}{b_{21}}$$

Therefore, for the filter $\frac{A(z_1, z_2)}{B(z_1, z_2)}$ the second condition of

Theorem 3 is verified if and only if $|\hat{z}_1| > 1$, that is

$$\left| - \frac{b_{11}}{b_{21}} \right| > 1$$

which is equivalent to (24).

Since (23) and (24) compare magnitudes only, the same conditions apply to $\frac{A(z_1, z_2)}{B(z_1, z_2)}$ and $\frac{A^*(z_1^*, z_2^*)}{B^*(z_1^*, z_2^*)}$, and therefore these filters are either both stable or both unstable. If both are stable, then by Proposition 1 in Appendix D, the filter given in (22) is stable.

Q.E.D.

Specializing Corollary 1 to rotated filters with their coefficients as defined in (21) we obtain Corollary 2.

Corollary 2: Rotating a stable one-dimensional continuous filter by an angle β in the (s_1, s_2) plane and applying the two-dimensional bilinear z-transform, the resulting two-dimensional digital filter is stable if $270^\circ < \beta < 360^\circ$

Proof: Substituting (21) into (23) and (24) we obtain (28) and (29) as sufficient stability conditions for rotated filters

$$\left| |a| - |\cos\beta| \right| > |a + \cos\beta| \quad (28)$$

$$\left| \cos\beta - \sin\beta - a \right| > \left| \cos\beta + \sin\beta - a \right| \quad (29)$$

where

$$a = \operatorname{Re}[\frac{T}{2} p] \quad \text{and } p \text{ represents the location of a pole.}$$

There are no values of a and β satisfying (28); at best, both sides of (28) are equal. This is because the transformation $B(z_1, z_2) = 0$ has a fixed point at $z_1 = z_2 = -1$. This pole is cancelled by a zero because the transformation $A(z_1, z_2) = 0$ has also a fixed point at $z_1 = z_2 = -1$. However, the pole and zero might not exactly cancel each

other because of quantization. To avoid this, since the centre of the circle image lies on the real axis (see the Corollary in Appendix C), we can always shift the circle image to the left by a small distance ϵ . This is illustrated in Fig. 6.

The set of values that satisfy (28) with equality is*

$$S_1 = \{a, \beta : (a < 0 \cap \cos \beta > 0) \cup (a > 0 \cap \cos \beta < 0)\}$$

The set of values that satisfy (29) is

$$S_2 = \{a, \beta : (\sin \beta > 0 \cap a > \cos \beta) \cup (\sin \beta < 0 \cap a < \cos \beta)\}$$

The set of values of a and β for which rotated filters are stable is given by the intersection of set S_1 and set S_2 .

$$S_1 \cap S_2 = \{a, \beta : (a < 0 \cap 270^\circ < \beta < 360^\circ) \cup (a > 0 \cap 90^\circ < \beta < 180^\circ)\}$$

The regions of stability are illustrated in Fig. 7.

Q.E.D.

Theorem 4 in Appendix A indicates that it might be possible to obtain stable filters for other angles of rotation by changing the direction of recursion. That is, by substituting either z_1 by z_1^{-1} , or z_2 by z_2^{-1} , or both, in the filter transfer function $H(z_1, z_2)$.

The application of Corollary 1 to all cases of interest results in Table V.

* \cap means "and" , \cup means "or"

Given a one-dimensional continuous filter whose transfer function $H(s)$ has poles either in the left half plane (L.H.P.) or in the right half plane (R.H.P.) and given an angle of rotation β from Table V we may obtain the transfer function of a two-dimensional digital filter which is guaranteed to be stable. It should be noted that changing z_1 to z_1^{-1} is equivalent to a rotation through a negative angle with respect to the f_1 axis (horizontal mirror image) and changing z_2 to z_2^{-1} is equivalent to a rotation through a negative angle with respect to the f_2 axis (vertical mirror image). It should be also noted that starting with a one-dimensional continuous filter either with poles in the L.H.P. and angle of rotation β or with poles in the R.H.P. and angle of rotation $\beta + 180^\circ$ results in a two-dimensional digital filter with the same magnitude response (cf. (21)).

Therefore, it can be shown that all the possibilities indicated by Table V result in a filter with the same magnitude response, that is, the effective angle of rotation always satisfies $270^\circ < \beta_e < 360^\circ$. To obtain a rotation through a different angle the data can be filtered in a distinct manner. Consider for example the following.

In the frequency domain the filtering operation can be represented as (cf. (9))

$$G(e^{-j\omega_1 T}, e^{-j\omega_2 T}) = H(e^{-j\omega_1 T}, e^{-j\omega_2 T}) \cdot F(e^{-j\omega_1 T}, e^{-j\omega_2 T}) \quad (30)$$

It is easy to see that a rotation of the filter transfer function $H(e^{-j\omega_1 T}, e^{-j\omega_2 T})$ by an angle θ is equivalent to a rotation of the two-dimensional Fourier transform of the two-dimensional input signal

$F(e^{-j\omega_1 T}, e^{-j\omega_2 T})$ by an angle $-\theta$ and a rotation of the output by an angle θ . In Appendix E it is shown that the rotation of the two-dimensional input signal results in the same rotation of its two-dimensional Fourier transform. Therefore, to obtain a filter rotated by an angle β , when $0^\circ < \beta < 270^\circ$, we may rotate the matrix containing the input data by -90° , -180° , or -270° , rotate the filter by β_f , where $270^\circ < \beta_f < 360^\circ$ for stability, perform the filtering operation and finally rotate the output matrix to the original position. Since the data is usually given in the form of a matrix, which allows us to perform the digital filtering operation, it can be easily rotated by angles which are multiples of 90° .

When choosing the angles of rotation for the data and for the filter the following equation has to be satisfied.

$$\beta_e = \beta_f - \beta_d \quad (31)$$

where

β_e = effective angle of rotation of the filter
with respect to the original data matrix.

β_f = angle of rotation of the filter
 $(270^\circ < \beta_f < 360^\circ$ for stability).

$\beta_d = k \cdot 90^\circ$ = angle of rotation of the input
data matrix (k is an integer)

Thus, given β_e , the appropriate value of k has to be chosen such that $\beta_f = \beta_e + k \cdot 90^\circ$ and $270^\circ < \beta_f < 360^\circ$ are satisfied. Note that

all the angles are measured counter-clockwise and modulo 360° , so $k \geq 0$ because $\beta_f \geq \beta_e$ always. After filtering, the output matrix has to be rotated by an angle $-\beta_d$. The current values of k and β_d follow

β_e	k	β_d
$0^\circ < \beta_e < 90^\circ$	3	270°
$90^\circ < \beta_e < 180^\circ$	2	180°
$180^\circ < \beta_e < 270^\circ$	1	90°
$270^\circ < \beta_e < 360^\circ$	0	0°

The rotation of the data is not the only possible solution.

There are other transformations of the data which change the direction of recursion. These transformations are the matrix transposition with respect to each diagonal and the horizontal and vertical mirror images. An alternative solution to the transformation of the data may be obtained by using algorithms recursing in other directions.

3. Design of Filters with Circular Symmetry

We have shown that stable rotated filters can be obtained for any angle of rotation(β_e). Therefore, cascading a number of rotated filters whose angles of rotation are uniformly distributed over 180°

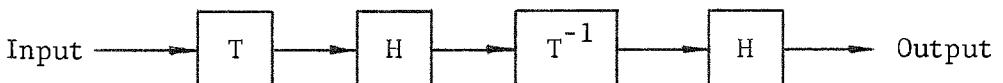
results in a magnitude response which approximates a circularly symmetric function by a polygonal. This polygonal has an even number of sides because with each rotated filter we obtain two sides of the polygonal. The more filters that are cascaded, the more sides the polygonal has and the better the circular symmetry.

Suppose that the angles of rotation are between 180° and 360° . When the angle of rotation is β , $180^\circ < \beta < 270^\circ$, we have to transform the data matrix according to what was said in IV.2, then filter with a $\beta + 90^\circ$ rotated filter, and inverse-transform the output matrix. If β is uniformly distributed between 180° and 360° , these filters rotated by $\beta + 90^\circ$, $180^\circ < \beta < 270^\circ$, coincide with those rotated by β , $270^\circ < \beta < 360^\circ$. Therefore, we need a filter consisting of rotated filters whose angles of rotation are distributed between 270° and 360° . The contour levels of the magnitude response of that filter have approximatively elliptical shapes with major axis oriented 315° . We refer to this class of filters as elliptical filters. Fig. 8 shows an example of an elliptical filter obtained by cascading the filters shown in Figs. 2, 3, and 4. To obtain a circularly symmetric magnitude response the following sequence of operations is used.

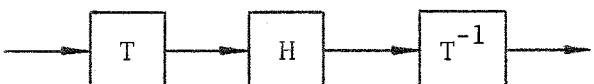


where the boxes represent linear operations, H denotes the elliptical filter, and T denotes the transformation (e.g., rotation by 90°). Here we assume that the impulse response of the elliptical filter

becomes negligible fast enough with respect to the dimensions of the data. An alternative solution is



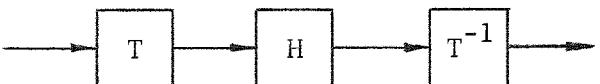
The combination



is equivalent to an elliptical filter oriented 225° . The magnitude response of this filter is shown in Fig. 9. Cascading the filters in Figs. 8 and 9 we obtain the filter shown in Fig. 12.

Figs. 10 through 15 show the magnitude responses obtained by cascading 2, 4, 6, 8, 10 and 12 rotated filters respectively. Figs. 16 through 21 show the same magnitude responses in one quadrant only. We observe that cascading more than four rotated filters the polygonal shape of the contour levels cannot be appreciated. Note also that as the number of rotated filters which are cascaded increases, the attenuation also increases and the cutoff region becomes steeper.

Tables 6 through 13 give the coefficients of the rotated filters from which Figs. 8 through 15, respectively, were obtained. The angle of rotation β , in degrees, is also given in the tables and the asterisk denotes an unstable filter. These unstable filters should be realized by implementing



where the filter H has been rotated by $\beta + 90^\circ$ and the transformation T is a rotation by 90° .

Two-dimensional filters with zero-phase response can be obtained [3]. Suppose for example that the following linear operation is performed



where H denotes an elliptical filter oriented 315° and T denotes a rotation by 90° . This is equivalent to a cascade of an even number of rotated filters whose angles of rotation are uniformly distributed between 0° and 360° . This filter has zero phase response. Indeed, its transfer function can be written as follows

$$\hat{H}(z_1, z_2) = H(z_1, z_2) H(z_1^{-1}, z_2^{-1})$$

where $H(z_1, z_2)$ is the transfer function of a cascade of rotated filters whose angles of rotation are distributed over 180° . Clearly, the filter $\hat{H}(z_1, z_2)$ has zero-phase response and its magnitude response is the square of the magnitude response of $H(z_1, z_2)$.

V. REALIZATION

1. Two-Dimensional Complex Cascade Programming

Algorithms for implementing two-dimensional recursive filters have been given by Shanks in [18]. The technique of complex cascade programming [16] in two-dimensions is developed here. This method is

suitable only for factorable filters such as those presented in Section IV. It has the advantage of reducing the effect of computational errors.

For complex cascade programming (20) is expressed in the form

$$H(z_1, z_2) = A \prod_{i=1}^r H_i^R(z_1, z_2) \prod_{j=r+1}^n H_j(z_1, z_2) \cdot H_j^*(z_1^*, z_2^*)$$

where $2n-r$ is the order of the filter, $H_i^R(z_1, z_2)$ for $0 < i \leq r$ are the filters in the cascade which have real coefficients, and $H_i(z_1, z_2)$ and $H_i^*(z_1^*, z_2^*)$ for $r < i \leq n$ are the remaining factors in the cascade. That is,

$$H_i^R(z_1, z_2) = \frac{a_{11}^i + a_{21}^i z_1 + a_{12}^i z_2 + a_{22}^i z_1 z_2}{1 + b_{21}^i z_1 + b_{12}^i z_2 + b_{22}^i z_1 z_2} \quad 0 < i \leq r \quad (33a)$$

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

$$H_i(z_1, z_2) = \frac{a_{11}^i + a_{21}^i z_1 + a_{12}^i z_2 + a_{22}^i z_1 z_2}{1 + b_{21}^i z_1 + b_{12}^i z_2 + b_{22}^i z_1 z_2} \quad r < i \leq n \quad (33b)$$

$$H_i^*(z_1^*, z_2^*) = \frac{(a_{11}^i)^* + (a_{21}^i)^* z_1 + (a_{12}^i)^* z_2 + (a_{22}^i)^* z_1 z_2}{1 + (b_{21}^i)^* z_1 + (b_{12}^i)^* z_2 + (b_{22}^i)^* z_1 z_2} \quad r < i \leq n \quad (33c)$$

Without loss of generality we have assumed that $b_{11}^i = 1$ for $0 < i \leq n$. If $b_{11}^i \neq 1$ for some i we can always divide the numerator

and the denominator of the corresponding filter of the cascade by b_{11}^i .

A block diagram for complex cascade programming is given in Fig. 22. With reference to this block diagram and (33), the complex cascade programming of $H(z_1, z_2)$ consists of successively solving the following difference equations for $g(k, \ell)$

$$d_o(k, \ell) = A f(k, \ell) \quad (34a)$$

$$\begin{aligned} d_i(k, \ell) = & a_{11}^i d_{i-1}(k, \ell) + a_{21}^i d_{i-1}(k-1, \ell) + \\ & + a_{12}^i d_{i-1}(k, \ell-1) + a_{22}^i d_{i-1}(k-1, \ell-1) + \\ & - b_{21}^i d_i(k-1, \ell) - b_{12}^i d_i(k, \ell-1) + \\ & - b_{22}^i d_i(k-1, \ell-1) \end{aligned}$$

$$\text{for } i = 1, 2, \dots, r \quad (34b)$$

$$\begin{aligned} \operatorname{Re}[e_i(k, \ell)] = & \operatorname{Re}[a_{11}^i] d_{i-1}(k, \ell) + \operatorname{Re}[a_{21}^i] d_{i-1}(k-1, \ell) + \\ & + \operatorname{Re}[a_{12}^i] d_{i-1}(k, \ell-1) + \operatorname{Re}[a_{22}^i] d_{i-1}(k-1, \ell-1) + \\ & - \operatorname{Re}[b_{21}^i] \operatorname{Re}[e_i(k-1, \ell)] + \operatorname{Im}[b_{21}^i] \operatorname{Im}[e_i(k-1, \ell)] + \\ & - \operatorname{Re}[b_{12}^i] \operatorname{Re}[e_i(k, \ell-1)] + \operatorname{Im}[b_{12}^i] \operatorname{Im}[e_i(k, \ell-1)] + \\ & - \operatorname{Re}[b_{22}^i] \operatorname{Re}[e_i(k-1, \ell-1)] + \operatorname{Im}[b_{22}^i] \operatorname{Im}[e_i(k-1, \ell-1)] \end{aligned} \quad (34c)$$

$$\begin{aligned}
 \text{Im}[e_i(k, \ell)] = & \text{Im}[a_{11}^i] d_{i-1}(k, \ell) + \text{Im}[a_{21}^i] d_{i-1}(k-1, \ell) + \\
 & + \text{Im}[a_{12}^i] d_{i-1}(k, \ell-1) + \text{Im}[a_{22}^i] d_{i-1}(k-1, \ell-1) + \\
 & - \text{Im}[b_{21}^i] \text{Re}[e_i(k-1, \ell)] - \text{Re}[b_{21}^i] \text{Im}[e_i(k-1, \ell)] + \\
 & - \text{Im}[b_{12}^i] \text{Re}[e_i(k, \ell-1)] - \text{Re}[b_{12}^i] \text{Im}[e_i(k, \ell-1)] + \\
 & - \text{Im}[b_{22}^i] \text{Re}[e_i(k-1, \ell-1)] - \text{Re}[b_{22}^i] \text{Im}[e_i(k-1, \ell-1)] \quad (34d)
 \end{aligned}$$

$$\begin{aligned}
 d_i(k, \ell) = & \text{Re}[a_{11}^i] \text{Re}[e_i(k, \ell)] + \text{Im}[a_{11}^i] \text{Im}[e_i(k, \ell)] + \\
 & + \text{Re}[a_{21}^i] \text{Re}[e_i(k-1, \ell)] + \text{Im}[a_{21}^i] \text{Im}[e_i(k-1, \ell)] + \\
 & + \text{Re}[a_{12}^i] \text{Re}[e_i(k, \ell-1)] + \text{Im}[a_{12}^i] \text{Im}[e_i(k, \ell-1)] + \\
 & + \text{Re}[a_{22}^i] \text{Re}[e_i(k-1, \ell-1)] + \text{Im}[a_{22}^i] \text{Im}[e_i(k-1, \ell-1)] + \\
 & - \text{Re}[b_{21}^i] d_i(k-1, \ell) - \text{Re}[b_{12}^i] d_i(k, \ell-1) + \\
 & - \text{Re}[b_{22}^i] d_i(k-1, \ell-1) \\
 \text{for } i = r+1, \dots, n \quad (34e)
 \end{aligned}$$

$$g(k, \ell) = d_n(k, \ell) \quad (34f)$$

It should be observed that the sequence $\{d_i(k, \ell) : i=0, 1, 2, \dots, n\}$ is real and the sequence $\{e_i(k, \ell) : i=r+1, r+2, \dots, n\}$ is complex. A

feature of the recursive equations (34) is that they involve real arithmetic operations only.

Examples of impulses responses normalized to a peak value of 1.0 are shown in Figs. 23 through 29. In every case equations (34) were solved for 101x101 discrete input values, where the only nonzero value was $f(51,51) = 1.0$. Figures 23, 24, and 25 show impulse responses in one quadrant only because they were obtained with causal filters. The impulse response shown in Fig. 24 was obtained with an unstable filter, a second-order Butterworth filter rotated 225° . In Figs. 26 to 29 the idea of combining elliptical filters and rotations by multiples of 90° was used, so the impulse response exists in more than one quadrant.

All the equations shown in this thesis were coded in FORTRAN IV. The most relevant programs are listed in Appendix G. The machine used for the computations was the IBM SYSTEM/370-165 with a FORTRAN IV (G, level 21) compiler and the contour maps were produced with a CALCOMP plotter. Table XIV gives the CPU time employed in obtaining the data for Figs. 23 through 29.

2. Computational Errors in Two-Dimensional Complex Cascade Programming

Since operations in a digital computer are carried out with only finite bit accuracy there is an accumulative roundoff error which propagates through the stages of the filter. In the steady state, this computational error can be represented by a set of additive noise processes at the output of an errorless computer [19]. We will find

an expression for the computational error to be added to the computer output of the ideal realization to account for the finite wordlength of the actual machine. This analysis is based on the one-dimensional analysis in [16].

If we call $w_i(k, \ell)$, $i=0, 1, 2, \dots, 3n-2r$, the roundoff errors incurred in the evaluation of equations (34), then the actual outputs of the elements in Fig. 22 become

$$d'_0(k, \ell) = A f(k, \ell) + w_0(k, \ell) \quad (35a)$$

$$\begin{aligned} d'_i(k, \ell) = & a_{11}^i d'_{i-1}(k, \ell) + a_{21}^i d'_{i-1}(k-1, \ell) + \\ & + a_{12}^i d'_{i-1}(k, \ell-1) + a_{22}^i d'_{i-1}(k-1, \ell-1) + \\ & - b_{21}^i d'_i(k-1, \ell) - b_{12}^i d'_i(k, \ell-1) + \\ & - b_{22}^i d'_i(k-1, \ell-1) + w_i(k, \ell) \end{aligned} \quad \text{for } i=1, 2, \dots, r \quad (35b)$$

$$\begin{aligned} \operatorname{Re}[e'_i(k, \ell)] = & \operatorname{Re}[a_{11}^i] d'_{i-1}(k, \ell) + \operatorname{Re}[a_{21}^i] d'_{i-1}(k-1, \ell) + \\ & + \operatorname{Re}[a_{12}^i] d'_{i-1}(k, \ell-1) + \operatorname{Re}[a_{22}^i] d'_{i-1}(k-1, \ell-1) + \\ & - \operatorname{Re}[b_{21}^i] \operatorname{Re}[e'_i(k-1, \ell)] + \operatorname{Im}[b_{21}^i] \operatorname{Im}[e'_i(k-1, \ell)] + \\ & - \operatorname{Re}[b_{12}^i] \operatorname{Re}[e'_i(k, \ell-1)] + \operatorname{Im}[b_{12}^i] \operatorname{Im}[e'_i(k, \ell-1)] + \\ & - \operatorname{Re}[b_{22}^i] \operatorname{Re}[e'_i(k-1, \ell-1)] + \operatorname{Im}[b_{22}^i] \operatorname{Im}[e'_i(k-1, \ell-1)] + \\ & + w_{3i-2r-2}(k, \ell) \end{aligned} \quad (35c)$$

$$\begin{aligned}
\operatorname{Im}[e_i^i(k, \ell)] = & \operatorname{Im}[a_{11}^i] d_{i-1}^i(k, \ell) + \operatorname{Im}[a_{21}^i] d_{i-1}^i(k-1, \ell) + \\
& + \operatorname{Im}[a_{12}^i] d_{i-1}^i(k, \ell-1) + \operatorname{Im}[a_{22}^i] d_{i-1}^i(k-1, \ell-1) + \\
& - \operatorname{Im}[b_{21}^i] \operatorname{Re}[e_i^i(k-1, \ell)] - \operatorname{Re}[b_{21}^i] \operatorname{Im}[e_i^i(k-1, \ell)] + \\
& - \operatorname{Im}[b_{12}^i] \operatorname{Re}[e_i^i(k, \ell-1)] - \operatorname{Re}[b_{12}^i] \operatorname{Im}[e_i^i(k, \ell-1)] + \\
& - \operatorname{Im}[b_{22}^i] \operatorname{Re}[e_i^i(k-1, \ell-1)] - \operatorname{Re}[b_{22}^i] \operatorname{Im}[e_i^i(k-1, \ell-1)] + \\
& + w_{3i-2r-1}(k, \ell)
\end{aligned} \tag{35d}$$

$$\begin{aligned}
d_i^i(k, \ell) = & \operatorname{Re}[a_{11}^i] \operatorname{Re}[e_i^i(k, \ell)] + \operatorname{Im}[a_{11}^i] \operatorname{Im}[e_i^i(k, \ell)] + \\
& + \operatorname{Re}[a_{21}^i] \operatorname{Re}[e_i^i(k-1, \ell)] + \operatorname{Im}[a_{21}^i] \operatorname{Im}[e_i^i(k-1, \ell)] + \\
& + \operatorname{Re}[a_{12}^i] \operatorname{Re}[e_i^i(k, \ell-1)] + \operatorname{Im}[a_{12}^i] \operatorname{Im}[e_i^i(k, \ell-1)] + \\
& + \operatorname{Re}[a_{22}^i] \operatorname{Re}[e_i^i(k-1, \ell-1)] + \operatorname{Im}[a_{22}^i] \operatorname{Im}[e_i^i(k-1, \ell-1)] + \\
& - \operatorname{Re}[b_{21}^i] d_i^i(k-1, \ell) - \operatorname{Re}[b_{12}^i] d_i^i(k, \ell-1) + \\
& - \operatorname{Re}[b_{22}^i] d_i^i(k-1, \ell-1) + w_{3i-2r}(k, \ell)
\end{aligned} \tag{35e}$$

for $i=r+1, r+2, \dots, n$

$$g^i(k, \ell) = d_n^i(k, \ell) \tag{35f}$$

Defining the computational error quantities as the difference between the actual and the ideal outputs of the elements in Fig. 22, we have

$$\varepsilon_o(k, \ell) \triangleq d_o^i(k, \ell) - d_o(k, \ell) \tag{36a}$$

$$\varepsilon_i(k, \ell) \triangleq d_i^i(k, \ell) - d_i(k, \ell) \quad 0 < i \leq r \tag{36b}$$

$$\varepsilon_{3i-2r-2}(k, \ell) \triangleq \operatorname{Re}[e_i^*(k, \ell)] - \operatorname{Re}[e_i(k, \ell)] \quad r < i \leq n \quad (36c)$$

$$\varepsilon_{3i-2r-1}(k, \ell) \triangleq \operatorname{Im}[e_i^*(k, \ell)] - \operatorname{Im}[e_i(k, \ell)] \quad r < i \leq n \quad (36d)$$

$$\varepsilon_{3i-2r}(k, \ell) \triangleq d_i^*(k, \ell) - d_i(k, \ell) \quad r < i \leq n \quad (36e)$$

then, from (34f), (35f), and (36e) the computational error $\varepsilon_c(k, \ell)$ on the computer output is evidently $\varepsilon_{3n-2r}(k, \ell)$ where

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

Taking the two-dimensional z-transform of equations (34) and (35) the block diagrams in Fig. 30 were obtained and the relationship existing between the computational-error sequence $\varepsilon_c(k, \ell)$ and the error sequences $\{w_i(k, \ell)\}$ was determined. This relation is shown in (37) and in Fig. 31.

$$\Sigma_c(z_1, z_2) = \Sigma_{3n-2r}(z_1, z_2) = \sum_{i=0}^{3n-2r} C_i(z_1, z_2) w_i(z_1, z_2) \quad (37)$$

where

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

$$C_i(z_1, z_2) = \frac{1}{1 + G_3^i} \left[\prod_{j=i+1}^r H_j(z_1, z_2) \right] \left[\prod_{j=r+1}^n H_j(z_1, z_2) H_j^*(z_1^*, z_2^*) \right]$$

for $0 < i \leq r$

$$C_{3i-2r-2}(z_1, z_2) = \frac{-G_4^i [G_1^i (1+G_3^i) - G_2^i G_4^i]}{[1+G_3^i] [(1+G_3^i)^2 - (G_4^i)^2]} \left[\prod_{j=i+1}^n H_j(z_1, z_2) H_j^*(z_1^*, z_2^*) \right]$$

for $r < i \leq n$

$$C_{3i-2r-1}(z_1, z_2) = \frac{G_2^i (1+G_3^i) + G_1^i G_4^i}{[1+G_3^i] [(1+G_3^i)^2 + (G_4^i)^2]} \left[\prod_{j=i+1}^n H_j(z_1, z_2) H_j^*(z_1^*, z_2^*) \right]$$

for $r < i \leq n$

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

$$G_1^i \triangleq G_1^i(z_1, z_2) \triangleq \operatorname{Re}[a_{11}^i] + \operatorname{Re}[a_{21}^i] z_1 + \operatorname{Re}[a_{12}^i] z_2 + \operatorname{Re}[a_{22}^i] z_1 z_2$$

$$G_2^i \triangleq G_2^i(z_1, z_2) \triangleq \operatorname{Im}[a_{11}^i] + \operatorname{Im}[a_{21}^i] z_1 + \operatorname{Im}[a_{12}^i] z_2 + \operatorname{Im}[a_{22}^i] z_1 z_2$$

$$G_3^i \triangleq G_3^i(z_1, z_2) \triangleq \operatorname{Re}[b_{21}^i] z_1 + \operatorname{Re}[b_{12}^i] z_2 + \operatorname{Re}[b_{22}^i] z_1 z_2$$

$$G_4^i \triangleq G_4^i(z_1, z_2) \triangleq \operatorname{Im}[b_{21}^i] z_1 + \operatorname{Im}[b_{12}^i] z_2 + \operatorname{Im}[b_{22}^i] z_1 z_2$$

for $i=1, 2, \dots, n$

The product term in the equations above is set to unity when $\min(j) \geq n$.

VI. CHOICE OF FILTER PARAMETERS

In Section IV.3 it was shown how the cutoff frequency and the shape of the magnitude response of a two-dimensional filter depend on the number of rotated filters being cascaded.

Since our goal is the design of filters with specified cutoff frequency and shape, in this section we study how to control them.

1. Cutoff frequency

A transformation which changes the cutoff frequency of a filter was the first thing which was thought of in order to compensate for the diminution in the cutoff frequency when a number of rotated filters are cascaded. This transformation, analogous to that used in one dimension to transform a low-pass filter into another low-pass filter (with

different cutoff frequency), was the following

$$z_1 \rightarrow \frac{z_1 - \alpha}{1 - \alpha z_1} \quad z_2 \rightarrow \frac{z_2 - \alpha}{1 - \alpha z_2} \quad (38)$$

where

$$\alpha = \frac{\sin(\frac{\omega_c - \omega_u}{2} T)}{\sin(\frac{\omega_c + \omega_u}{2} T)} \quad (39)$$

ω_c = cutoff frequency of the filter

ω_u = desired cutoff frequency

T = sampling period

Applying this transformation to the transfer function of a filter, such as that given in (20), we obtain

$$\hat{H}(z_1, z_2) = A \prod_{i=1}^M \frac{\hat{a}_{11}^i + \hat{a}_{21}^i z_1 + \hat{a}_{12}^i z_2 + \hat{a}_{22}^i z_1 z_2}{\hat{b}_{11}^i + \hat{b}_{21}^i z_1 + \hat{b}_{12}^i z_2 + \hat{b}_{22}^i z_1 z_2} \quad (40)$$

where

$$\hat{a}_{11}^i = a_{11}^i - \alpha (a_{21}^i + a_{12}^i) + \alpha^2 a_{22}^i \quad (41a)$$

$$\hat{a}_{21}^i = a_{21}^i - \alpha (a_{11}^i + a_{22}^i) + \alpha^2 a_{12}^i \quad (41b)$$

$$\hat{a}_{12}^i = a_{12}^i - \alpha (a_{11}^i + a_{22}^i) + \alpha^2 a_{21}^i \quad (41c)$$

$$\hat{a}_{22}^i = a_{22}^i - \alpha (a_{21}^i + a_{12}^i) + \alpha^2 a_{11}^i \quad (41d)$$

$$\hat{b}_{11}^i = b_{11}^i - \alpha (b_{21}^i + b_{12}^i) + \alpha^2 b_{22}^i \quad (41e)$$

$$\hat{b}_{21}^i = b_{21}^i - \alpha (b_{11}^i + b_{22}^i) + \alpha^2 b_{12}^i \quad (41f)$$

$$\hat{b}_{12}^i = b_{12}^i - \alpha (b_{11}^i + b_{22}^i) + \alpha^2 b_{21}^i \quad (41g)$$

$$\hat{b}_{22}^i = b_{22}^i - \alpha (b_{21}^i + b_{12}^i) + \alpha^2 b_{11}^i \quad (41h)$$

A program was written in FORTRAN IV to implement the transformation above. The principal features of that program follow.

- a) Given the coefficients of a two-dimensional digital filter (low-pass), find the cutoff frequency in a given direction (e.g., 0°).
- b) Determine α according to (39) and calculate the coefficients of the new filter according to equations (41).
- c) Find the cutoff frequency of the filter in a given direction (e.g., 0°).
- d) Check for stability.
- e) Find the magnitude response and plot a contour map.

The program was tested transforming the filter whose magnitude response is shown in Fig. 12. The cutoff of the original one-dimensional filter was 0.2, but after cascading the six rotated filters the cutoff frequency (cf. Fig. 12) was 0.150. The transformation was made to get a desired cutoff frequency of 0.350 but it resulted to be 0.3575. The new two-dimensional filter was unstable and the magnitude response became distorted. If the change in the cutoff frequency would have been smaller the results might have been better. No attempt was made to improve the method because another technique more accurate was found.

This technique consists of an iteration that modifies the cutoff frequency of the original one-dimensional continuous filter in the proper amount and sense until the desired cutoff frequency of the two-dimensional digital filter is reached.

This method was implemented by writing a program in FORTRAN IV (given in Appendix G) featuring the following steps.

a) Read in the coefficients of the one-dimensional continuous filter (with cutoff frequency normalized to unity), the number of rotations desired, and the desired cutoff frequency f_u of the two-dimensional digital filter given as a fraction of the Nyquist frequency.

b) Determine the coefficients of a two-dimensional digital filter with cutoff frequency f_u (using equations (21)).

c) Find the cutoff frequency f_c of the filter obtained in a given direction.

d) Determine the coefficients of a two-dimensional digital filter with cutoff frequency $2f_u - f_c$.

e) Repeat steps c) and d) until $|f_u - f_c| < \epsilon$ where ϵ is the specified maximum error of the cutoff frequency in that direction.

f) Determine the shape factors (see Section VI.2).

g) Check for stability.

h) Evaluate the magnitude response and plot a contour map.

The program described above was used to design many filters of different shapes and cutoff frequencies. The convergence was obtained quickly. The number of iterations required increases with

the cutoff frequency and decreases with the order of the original one-dimensional filter.

The following examples show typical iterations. In every case the specified maximum error of the cutoff frequency in the direction 0° was $\epsilon=0.000001$.

Example 1. Cascading four rotated second-order Butterworth filters design a two-dimensional recursive filter with cutoff frequency $f_u=0.35$.

f_u	f_c
0.3500000	0.2810633
0.4189366	0.3277398
0.4411968	0.3420500
0.4491468	0.3470686
0.4520782	0.3489091
0.4531690	0.3495914
0.4535776	0.3498468
0.4537308	0.3499420
0.4537887	0.3499787
0.4538100	0.3499917
0.4538182	0.3499967
0.4538215	0.3499993

Example 2. Cascading two rotated sixth-order Butterworth filters design a two-dimensional recursive filter with cutoff frequency $f_u=0.1$.

f_u	f_c
0.1000000	0.1295867
0.0704132	0.0918840
0.0785291	0.1023031
0.0762260	0.0993515
0.0768744	0.1001834
0.0766910	0.0999478
0.0767431	0.1000154
0.0767277	0.0999946
0.0767331	0.1000001
0.0767314	0.1000000

Example 3. Cascading eight rotated ninth-order Butterworth filters design a two-dimensional recursive filter with cutoff frequency $f_u = 0.1$.

f_u	f_c
0.1000000	0.0959838
0.1040161	0.0997757
0.1042403	0.0999878
0.1042524	0.1000000

2. Shape factors

Here, rather than giving huge tables of shape factors for a large class of filters, we describe some different kinds of shape factors that are suitable for two-dimensional digital filters. We also give some tables containing shape factors of two-dimensional

recursive filters derived from Butterworth filters of different orders.

Since the shape of the magnitude response of a filter depends on the cutoff frequency, as shown in Fig. 32, all these tables are given for filters normalized to a cutoff frequency of 0.1.

The first shape factor is the ratio in dB of the magnitude response at the cutoff frequency and the magnitude response at a frequency at a distance d beyond the cutoff frequency in a specified direction. This shape factor, that we call shape factor number 1, is given in Table XV for $d=0.05$ and $d=0.1$ in the directions 0° and 45° .

The second shape factor (No. 2) is the distance between the cutoff frequency and the frequency corresponding to an attenuation of A dB. Table XVI gives this shape factor for $A=10$, $A=20$, and $A=40$ in the directions 0° and 45° .

The third shape factor (No. 3) concerns the circular symmetry only and gives the area between the -3 dB contour and a circle of radius the cutoff frequency. These areas are given in Table XVII.

The fourth shape factor (No. 4) is both a measure of the steepness of the magnitude response and a measure of the circular symmetry. It gives the volume between the magnitude response of the filter and the magnitude response of an ideal filter (low-pass). The magnitude response of an ideal low-pass two-dimensional filter in polar coordinates is

$$H(v) \begin{cases} =1 & \text{for } v < f_c \\ =0 & \text{otherwise} \end{cases}$$

VII. CONCLUSIONS

The work described above concerns the design of stable two-dimensional recursive filters whose magnitude response approximates a circularly symmetric function. The principal advantages of the method are

- a) Stability can be guaranteed.
- b) The method leads to a recursive realization which is more efficient than a nonrecursive one.
- c) The filters are factorable into lower order filters, so that they can be efficiently realized. In general this factorization is not possible for two-dimensional recursive filters.
- d) The calculation of the coefficients of a filter using a digital computer is fast.

For the realization of these filters the technique of complex cascade programming has been developed in two dimensions and the associated computational errors have been analyzed. This technique has the advantage of minimizing the computational errors.

Finally, we have shown how to obtain the desired cutoff frequency in a given direction with a specified maximum error in that direction. We have also given tables of shape factors to help the designer choosing the order of the original filter and the number of rotations for a particular application.

Future work could be done on frequency domain synthesis in order to have more control of the error from the ideal characteristics of the filter. In particular, a problem of interest is to choose a

direction for the exact cutoff frequency such that the deviation from the cutoff frequency in all other directions is minimized (i.e., minimize shape factor number 3, see Section VI.2). The suitability of the filters discussed in this thesis for filtering images (or other practical applications) could be tested.

APPENDIX A

The following theorems concern stability of two-dimensional recursive digital filters.

Theorem 1 [11] : A two-dimensional filter is said to be stable (in the sense that a bounded input produces a bounded output) if and only if its impulse response satisfies the constraint

$$\sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} |h(m,n)| < \infty \quad (A.1)$$

Proof: The sufficiency is easily verified by considering as input to the filter a bounded sequence, and determining the output.

Consider the input $|x(m,n)| < M$, then by (2) the absolute value of the output is

$$|y(m,n)| = \left| \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} h(i,j) x(m-i, n-j) \right|$$

$$\leq M \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} |h(i,j)| < \infty$$

Hence, if $\sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} |h(i,j)| < \infty$, then $|y(m,n)| < 0$ which shows that $y(m,n)$ is bounded.

The necessity can be proved as follows. If (A.1) is not satisfied, then a bounded input can be found such that the output is

A.2

unbounded. Such an input is the sequence

$$x(m,n) = \begin{cases} +1 & \text{all } (m,n) \text{ such that } h(-m, -n) \geq 0 \\ -1 & \text{all other } (m,n) \end{cases}$$

For this sequence, the output at $m = n = 0$ is

$$\begin{aligned} y(0,0) &= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} h(i,j) x(m-i, n-j) \Big|_{m=n=0} = \\ &= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} |h(i,j)| = \infty \end{aligned}$$

Therefore, (A.1) is a necessary and sufficient condition for stability.

Q.E.D.

Theorem 2 [15]: A causal recursive filter with transfer function

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

where A and B are polynomials in z_1 and z_2 , is stable if and only if there are no values of z_1 and z_2 such that $B(z_1, z_2) = 0$, $|z_1| \leq 1$ and $|z_2| \leq 1$.

Proof: Let

$$H(z_1, z_2) = \frac{A(z_1, z_2)}{B(z_1, z_2)} = \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} h(i, j) z_1^{i-1} z_2^{j-1}. \quad (\text{A.2})$$

We want to show that

$$\sum_{i=1}^{\infty} \sum_{j=1}^{\infty} |h(i, j)| < \infty$$

if and only if $H(z_1, z_2)$ is analytic in the region

$$D = \{(z_1, z_2) : |z_1| \leq 1 \cap |z_2| \leq 1\}$$

Necessity: If $H(z_1, z_2)$ is analytic in D , we can find an $\epsilon > 0$

such that $H(z_1, z_2)$ is analytic in

$$D_1 = \{(z_1, z_2) : |z_1| < 1 + \epsilon \cap |z_2| < 1 + \epsilon\}$$

which implies that

$$\sum_{i=1}^{\infty} \sum_{j=1}^{\infty} h(i, j) z_1^i z_2^j$$

is absolutely convergent in D_1 . Therefore,

$$\sum_{i=1}^{\infty} \sum_{j=1}^{\infty} |h(i, j)| < \infty.$$

Sufficiency: If

$$\sum_{i=1}^{\infty} \sum_{j=1}^{\infty} |h(i, j)| < \infty,$$

then by the M test,

$$\sum_{i=1}^{\infty} \sum_{j=1}^{\infty} h(i, j) z_1^i z_2^j$$

is absolutely convergent in D , which implies in turn that $H(z_1, z_2)$ is analytic in D .

Q.E.D.

Theorem 3 [15]: A causal recursive filter with transfer function $H(z_1, z_2) = A(z_1, z_2)/B(z_1, z_2)$, where A and B are polynomials is stable if and only if:

- 1) the map of the unit circle of the z_1 plane, $\{z_1: |z_1| = 1\}$ into the z_2 plane, according to the equation $B(z_1, z_2) = 0$, lies outside the unit disk in the z_2 plane, $\{z_2: |z_2| \leq 1\}$; and
- 2) no point in the unit disk of the z_1 plane $\{z_1: |z_1| \leq 1\}$ maps into the point $z_2 = 0$ by the relation $B(z_1, z_2) = 0$.

Proof: We want to establish that the stability conditions of Theorems 2 and 3 are equivalent. It is obvious that the stability conditions of Theorem 2 imply those of Theorem 3. So we proceed to show the implication in the reverse order.

The two-variable polynomial $B(z_1, z_2) = 0$ defines an algebraic function $z_2 = f(z_1)$. We first modify the unit-circle contour in the z_1 plane to exclude any singular points of f inside the contour, resulting in a modified contour $\partial d'_1$, as shown in Fig. A.1. We use d'_1 to denote the closed region enclosed by $\partial d'_1$. A point $z_1 = z_1^0$ is called a singular point of $z_2 = f(z_1)$, if $B(z_1^0, z_2) = 0$, considered as an equation in z_2 , has multiple (finite or infinite) roots.

According to the theory of algebraic functions, in d'_1 the function $z_2 = f(z_1)$ has a number of branches, each of which is holomorphic. Therefore, from the maximum-modulus theorem, the maximum of $|f(z_1)|$ over d'_1 occurs on $\partial d'_1$, and the minimum of $f(z_1)$ over d'_1 can occur in the interior only if the minimum is zero. However, condition 2) of Theorem 3 says $f(z_1)$ is never zero in d'_1 . Therefore, the minimum of $f(z_1)$ occurs

on $\partial d_1'$, that is,

$$|f(d_1')| \geq \min |f(\partial d_1')|$$

which implies that if $|f(\partial d_1')| > 1$, then $|f(d_1')| > 1$, that is, to ensure that $f(d_1')$ lies outside the unit disk $d_2 = \{z_2 : |z_2| \leq 1\}$, it is sufficient to ensure that $f(\partial d_1')$ lies outside d_2 .

We are almost there, but not quite. What we really want to show is that if $|f(\partial d_1)| > 1$, then $|f(d_1)| > 1$, where $d_1 \equiv \{z_1 : |z_1| \leq 1\}$ and $\partial d_1 \equiv \{z_1 : |z_1| = 1\}$. Since the detour in $\partial d_1'$ can be any path leading from ∂d_1 to the singular point, what is left to show is simply that $|f(s)| > 1$, where s is the singular point. But since each branch of $z_2 = f(z_1)$ is continuous at $z_1 = s$, and since $|f(s + \epsilon e^{j\theta})| > 1$ for arbitrarily small ϵ and any θ , we have $|f(s)| > 1$.

Q.E.D.

Remark: Testing stability using these theorems is hard because for each particular value of $z_1 = \hat{z}_1$, where \hat{z}_1 belongs to the unit disk in Theorem 2 or it belongs to the unit circle in Theorem 3, the equation $B(\hat{z}_1, z_2) = 0$ has to be solved for z_2 . This has to be done for all (in practice a large number) of points in the unit disk (Theorem 2) or in the unit circle (Theorem 3).

Theorem 4 [15]: Among the four recursive filters which can be associated with $H(z_1, z_2) = A(z_1, z_2)/B(z_1, z_2)$, that is, $H(z_1, z_2)$ itself, $H(z_1^{-1}, z_2)$, $H(z_1, z_2^{-1})$, and $H(z_1^{-1}, z_2^{-1})$, at most one is stable.

Proof: It can easily be proved, using Theorem 2, that the filters $H(z_1, z_2)$, $H(z_1^{-1}, z_2)$, $H(z_1, z_2^{-1})$, and $H(z_1^{-1}, z_2^{-1})$ are stable if and only if there are no values of z_1 and z_2 such that $B(z_1, z_2) = 0$ and

$$1) \quad |z_1| \leq 1 \text{ and } |z_2| \leq 1 \quad \text{for } H(z_1, z_2)$$

$$2) \quad |z_1| \geq 1 \text{ and } |z_2| \leq 1 \quad \text{for } H(z_1^{-1}, z_2)$$

$$3) \quad |z_1| \leq 1 \text{ and } |z_2| \geq 1 \quad \text{for } H(z_1, z_2^{-1})$$

$$4) \quad |z_1| \geq 1 \text{ and } |z_2| \geq 1 \quad \text{for } H(z_1^{-1}, z_2^{-1})$$

It is evident that at most one of these four stability conditions, corresponding to the four recursing directions, can be satisfied.

APPENDIX B

The following proposition is essentially the convolution property of the two-dimensional z-transform.

Proposition: The two-dimensional z-transform of (B.1) is (B.2)

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

$$B(z_1, z_2) G(z_1, z_2) = A(z_1, z_2) F(z_1, z_2) \quad (\text{B.2})$$

where

$$A(z_1, z_2) \stackrel{\Delta}{=} \sum_{i=1}^{M_a} \sum_{j=1}^{N_a} a_{ij} z_1^{i-1} z_2^{j-1} \quad (\text{B.3a})$$

$$B(z_1, z_2) \stackrel{\Delta}{=} \sum_{i=1}^{M_b} \sum_{j=1}^{N_b} b_{ij} z_1^{i-1} z_2^{j-1} \quad (\text{B.3b})$$

$$F(z_1, z_2) \stackrel{\Delta}{=} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(m, n) z_1^m z_2^n \quad (\text{B.3c})$$

$$G(z_1, z_2) \stackrel{\Delta}{=} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} g(m, n) z_1^m z_2^n \quad (\text{B.3d})$$

Proof: Taking the two-dimensional z-transform as defined in (9) of both sides of (B.1), we obtain

$$\sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \sum_{i=1}^{M_b} \sum_{j=1}^{N_b} b_{ij} g(m-i+1, n-j+1) z_1^m z_2^n =$$

$$= \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \sum_{i=1}^{M_a} \sum_{j=1}^{N_a} a_{ij} f(m-i+1, n-j+1) z_1^m z_2^n$$

Inversion of the order of summation gives

$$\sum_{i=1}^{M_b} \sum_{j=1}^{N_b} b_{ij} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} g(m-i+1, n-j+1) z_1^m z_2^n = \\ = \sum_{i=1}^{M_a} \sum_{j=1}^{N_a} a_{ij} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(m-i+1, n-j+1) z_1^m z_2^n$$

Since z_1 and z_2 are unit delay operators we have

$$\sum_{i=1}^{M_b} \sum_{j=1}^{N_b} b_{ij} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} g(m, n) z_1^{m+i-1} z_2^{n+j-1} = \\ = \sum_{i=1}^{M_a} \sum_{j=1}^{N_a} a_{ij} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(m, n) z_1^{m+i-1} z_2^{n+j-1}$$

Now we can separate the summations in i and j from the summations in m and n obtaining

$$\sum_{i=1}^{M_b} \sum_{j=1}^{N_b} b_{ij} z_1^{i-1} z_2^{j-1} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} g(m, n) z_1^m z_2^n = \\ = \sum_{i=1}^{M_a} \sum_{j=1}^{N_a} a_{ij} z_1^{i-1} z_2^{j-1} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(m, n) z_1^m z_2^n$$

This equation is actually (B.2) with $A(z_1, z_2)$, $B(z_1, z_2)$, $F(z_1, z_2)$, and $G(z_1, z_2)$ as defined in (B.3)

Q.E.D.

APPENDIX C

Lemma 1: The image of the unit circle $\{z_1: |z_1| = 1\}$ by a bilinear transformation of the type

$$z_2 = - \frac{b_{11} + b_{21}}{b_{12} + b_{22}} z_1, \quad (\text{C.1})$$

where b_{11} , b_{21} , b_{12} , and b_{22} are any complex constants, is another circle in the z_2 plane of centre c and radius r given by

$$c = \frac{b_{21} b_{22}^* - b_{11} b_{12}^*}{|b_{12}|^2 - |b_{22}|^2}, \quad (\text{C.2})$$

$$r = \left| \frac{b_{11} b_{22} - b_{21} b_{12}}{|b_{12}|^2 - |b_{22}|^2} \right| \quad (\text{C.3})$$

Proof: The proof that (C.1) maps circles into circles can be found in any text on complex variables (see for example [17]) and will not be included here.

To find the centre and the radius of the circle image we have to determine two points z_{1A} and z_{1B} over the unit circle $\{z_1: |z_1| = 1\}$ such that their images z_{2A} and z_{2B} are diametrically opposite (i.e.,

C.2

maximally far apart) over the circle image. Under these circumstances

$$c = \frac{z_{2A} + z_{2B}}{2} \quad (\text{C.4})$$

$$r = \left| \frac{z_{2A} - z_{2B}}{2} \right| \quad (\text{C.5})$$

In order to see how we choose z_{1A} and z_{1B} we write (C.1) as follows

$$z_2 = -\frac{\frac{b_{11} + b_{21}}{b_{12} + b_{22}} z_1}{-\frac{b_{21}}{b_{22}}} = -\frac{\frac{b_{21}}{b_{22}} - \frac{b_{11} - b_{12}}{b_{12} + b_{22}} z_1}{z_1} = \mu + \frac{v}{\frac{b_{12} + b_{22}}{b_{12} + b_{22}} z_1} \quad (\text{C.6})$$

where

$$\mu = -\frac{b_{21}}{b_{22}}$$

$$v = -b_{11} + \frac{b_{21}}{b_{22}} b_{12}$$

The transformation (C.6) can be considered as a combination

of the following transformations

$$u = b_{12} + b_{22} z_1$$

$$v = \frac{1}{u}$$

$$z_2 = u + v v$$

namely, the product of a similarity, an inversion, and another similarity.

The similarity has the property of mapping straight lines into straight lines and so does the inversion $v = 1/u$ for those straight lines crossing the origin in the u plane. Since our domain of interest is $\{z_1 : |z_1| = 1\}$, the points in the u plane lying on a straight line crossing the origin, and that, according to the transformation

$u = b_{12} + b_{22} z_1$, are images of some z_1 such that $|z_1| = 1$, are the following (see Fig. C.1)

$$u_A = b_{12} - \left| b_{22} \right| \frac{b_{12}}{\left| b_{12} \right|} \quad (\text{C.7})$$

$$u_B = b_{12} + \left| b_{22} \right| \frac{b_{12}}{\left| b_{12} \right|} \quad (\text{C.8})$$

Therefore, by the transformation

$$z_2 = -\frac{b_{21}}{b_{22}} - (b_{11} - \frac{b_{21}}{b_{22}} b_{12}) \frac{1}{u}$$

the points z_{2A} and z_{2B} are

$$z_{2A} \stackrel{\Delta}{=} z_2 \Big|_{u=u_A} = -\frac{b_{21}}{b_{22}} - \frac{b_{11} b_{22} - b_{21} b_{12}}{b_{22}} \frac{|b_{12}|}{b_{12} (|b_{12}| - |b_{22}|)} \quad (C.9)$$

$$z_{2B} \stackrel{\Delta}{=} z_2 \Big|_{u=u_B} = -\frac{b_{21}}{b_{22}} - \frac{b_{11} b_{22} - b_{21} b_{12}}{b_{22}} \frac{|b_{12}|}{b_{12} (|b_{12}| + |b_{22}|)} \quad (C.10)$$

Substituting (C.9) and (C.10) into (C.4) and (C.5) the centre

and the radius are

$$c = \frac{z_{2A} + z_{2B}}{2} = \frac{b_{21} b_{22}^* - b_{11} b_{12}^*}{|b_{12}|^2 - |b_{22}|^2}$$

$$r = \left| \frac{z_{2A} - z_{2B}}{2} \right| = \frac{|b_{11} b_{22} - b_{21} b_{12}|}{|b_{12}|^2 - |b_{22}|^2}$$

Q.E.D.

Corollary: If the coefficients b_{11} , b_{21} , b_{12} , and b_{22} of (C.1) are defined by (21) the centre c and the radius r of the circle image are given by

$$c = - \frac{a}{a + \cos\beta} \quad (\text{C.11})$$

$$r = \frac{\cos\beta}{a + \cos\beta} \quad (\text{C.12})$$

where $a = \operatorname{Re} [\frac{T}{2} \mathbf{p}]$

Remarks: Note that the centre given by (C.11) is real, that is, it is going to lie always on the real axis. If $a + \cos\beta = 0$ the circle image becomes a straight line perpendicular to the real axis at the point $(-1, 0)$.

APPENDIX D

The following proposition is used in the proof of Corollary 1 in Section IV.

Proposition 1: A cascade of stable two-dimensional discrete filters results in a stable filter.

Proof: Let $h_1(m,n)$ and $h_2(m,n)$ be the impulse responses of the filters with transfer functions $H_1(z_1, z_2)$ and $H_2(z_1, z_2)$, respectively. The transfer function, $H(z_1, z_2)$, of these two filters in cascade is the product of their transfer functions.

$$H(z_1, z_2) = H_1(z_1, z_2) \cdot H_2(z_1, z_2)$$

Since multiplication in the z-domain results in convolution in the space domain (see Appendix B) the impulse response of the filter with transfer function $H(z_1, z_2)$ is

$$h(m,n) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} h_1(i,j) \cdot h_2(m-i, n-j) \quad (D.1)$$

We have to prove that if (cf. (3))

$$\sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} |h_1(m,n)| < \infty \quad (D.2)$$

and

$$\sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} |h_2(m,n)| < \infty \quad (D.3)$$

then

$$\sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} |h(m,n)| < \infty \quad (\text{D.4})$$

Indeed, from (D.1)

$$\begin{aligned} & \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} |h(m,n)| = \\ &= \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \left| \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} h_1(i,j) \cdot h_2(m-i, n-j) \right| \quad (\text{D.5}) \\ &\leq \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} |h_1(i,j)| \cdot |h_2(m-i, n-j)| = \\ &= \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} |h_1(i,j)| \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} |h_2(m-i, n-j)| < \infty \end{aligned}$$

Clearly, this proof can be iterated for any number of filters in cascade.

Q.E.D.

APPENDIX E

In this appendix we show that a rotation of a function in the space domain causes an equal rotation of its Fourier transform in the frequency domain.

Proposition: Let $f(x,y)$ be a function of two variables.

Its Fourier transform is defined by (E.1)

$$F(\omega_1, \omega_2) = \iint_{-\infty}^{\infty} f(x,y) \exp [-j(\omega_1 x + \omega_2 y)] dx dy \quad (\text{E.1})$$

If the function $f(x,y)$ is rotated by an angle θ , then its Fourier transform $F(\omega_1, \omega_2)$ is rotated by the same angle θ .

Proof: Rotating the function $f(x,y)$ by an angle θ is equivalent to rotating the axes x and y by an angle $-\theta$. The old coordinates (x,y) and the new coordinates (x',y') are related by the following equations.

$$x' = x \cos \theta - y \sin \theta \quad (\text{E.2a})$$

$$y' = x \sin \theta + y \cos \theta \quad (\text{E.2b})$$

The Jacobian of this transformation is equal to 1 and the

equations of the inverse transformation are

$$x = x' \cos \theta + y' \sin \theta \quad (\text{E.3a})$$

$$y = -x' \sin \theta + y' \cos \theta \quad (\text{E.3b})$$

Therefore, the rotated function is

$$f_R(x, y) = f(x', y') = f(x \cos \theta + y \sin \theta, x \sin \theta + y \cos \theta) \quad (\text{E.4})$$

Taking the Fourier transform of (E.4) and substituting the variables of integration according to the transformation in (E.3) we obtain

$$\begin{aligned} F_R(\omega_1, \omega_2) &= \iint_{-\infty}^{\infty} f_R(x, y) \cdot \exp[-j(\omega_1 x + \omega_2 y)] dx dy = \\ &= \iint_{-\infty}^{\infty} f(x \cos \theta - y \sin \theta, x \sin \theta + y \cos \theta) \cdot \\ &\quad \cdot \exp[-j(\omega_1 x + \omega_2 y)] dx dy = \\ &= \iint_{-\infty}^{\infty} f(x', y') \cdot \exp[-j(\omega'_1 x' + \omega'_2 y')] dx' dy' = \\ &= F(\omega'_1, \omega'_2) \end{aligned} \quad (\text{E.5})$$

where

$$\omega'_1 = \omega_1 \cos \theta - \omega_2 \sin \theta$$

$$\omega'_2 = \omega_1 \sin \theta + \omega_2 \cos \theta$$

These are the equations of a rotation of the frequency axes by an angle $-\theta$. Therefore the Fourier transform $F(\omega_1, \omega_2)$ has been rotated by an angle θ .

Q.E.D.

APPENDIX G

Program Listing

```
***** MAIN PROGRAM *****
C
C      COMPLEX CASCADE PROGRAMMING OF A TWO-DIMENSIONAL
C      RECURSIVE FILTER USING REAL ARITHMETIC.
C
REAL GET(5)/0.9,0.6,0.3,0.0,-0.3/
DATA NU/5/
INTEGER TIME(5,40)
LOGICAL NIR
REAL XA11( 6 ),XA21( 6 ),XA12( 6 ),XA22( 6 ),
-      YA11( 6 ),YA21( 6 ),YA12( 6 ),YA22( 6 ),
-      XB11( 6 ),XB21( 6 ),XB12( 6 ),XB22( 6 ),
-      YB11( 6 ),YB21( 6 ),YB12( 6 ),YB22( 6 ),
-      X1(102, 6 ),X2(102, 6 ),Y1(102, 6 ),Y2(102, 6 ),
-      D1(102, 6 ),D2(102, 6 ),D(102,102)
CALL ERRSET (207,256,0,1)
C
C      INITIALIZATION
NX=102
NY=102
ITIME=0
C
C      INITIALIZE FOR CONTOURING
I1=2
J1=2
IPAG=1
XSIZE=6.0
YSIZE=6.0
XSIZE2=XSIZE*0.5
YSIZE2=YSIZE*0.5
IPLOT=0
NDEC=10
IN=0
X=2.0
Y=3.2
C
1 READ(5,5000) NP, MD, ND, INORM
IF(NP.NE.0) GO TO 5
C
C      END OF EXECUTION
IF(IPLOT.NE.7) CALL PLOTND
WRITE(6,6010) ((TIME(K,L),K=1,5),L,L=1,ITIME)
```

```

STOP
C
C      IMPULSE RESPONSE OR INPUT DATA?
5 CONTINUE
  ITIME=ITIME+1
  CALL UTTIME (TIME(1,ITIME))
  NIR=.TRUE.
  IF(MD.GT.0 .AND. ND.GT.0) GO TO 15
  MD=IABS(MD)
  ND=IABS(ND)
  NIR=.FALSE.
15 KEND=MD+1
  LEND=ND+1
  IEND=NP+1
C
C      READ COEFFICIENTS OF THE TWO-DIMENSIONAL FILTER
C      (DO NOT READ COMPLEX CONJUGATES)
25 READ(5,5001)
  -(XA11(I),YA11(I),XA21(I),YA21(I),XA12(I),YA12(I),XA22(I),YA22(I),
  - I=2,IEND)
  READ(5,5001)
  -(XB11(I),YB11(I),XB21(I),YB21(I),XB12(I),YB12(I),XB22(I),YB22(I),
  - I=2,IEND)
C
C      GENERATE IMPULSE OR READ INPUT DATA MATRIX
  DO 6 K=1,KEND
  DO 6 L=1,LEND
  6 D(K,L)=0.0
  D(52,52)=1.0
35 IF(NIR) READ(5,5002) ((D(K,L),L=2,LEND),K=2,KEND)
  ITIME=ITIME+1
  CALL UTTIME (TIME(1,ITIME))

C      DO 700 IC=1,4

C      SET THE BOUNDARY CONDITIONS
40 DO 60 I=1,IEND
  D2(1,I)=0.0
  X2(1,I)=0.0
  Y2(1,I)=0.0
  DO 60 K=1,KEND
  D1(K,I)=0.0
  X1(K,I)=0.0
60 Y1(K,I)=0.0
C
C      RECURSE BY ROWS
  DO 666 L=2,LEND
  L1=L-1
C
C      RECURSE BY COLUMNS

```

```

DO 606 K=2,KEND
K1=K-1
D2(K,1)=D(K,L)
C
C   CASCADE OF BILINEAR FILTERS
DO 600 I=2,IEND
I1=I-1
IF(YA11(I).NE.0.0 .OR. YB12(I).NE.0.0) GO TO 200
D2(K,I) = XA11(I)* D2(K ,I1)+XA21(I)* D2(K1,I1)
-      +XA12(I)* D1(K ,I1)+XA22(I)* D1(K1,I1)
-      -XB21(I)* D2(K1,I )-XB12(I)* D1(K ,I )
-      -XB22(I)* D1(K1,I )
GO TO 600
200 X2(K,I) = XA11(I)* D2(K ,I1)+XA21(I)* D2(K1,I1)
-      +XA12(I)* D1(K ,I1)+XA22(I)* D1(K1,I1)
-      -XB21(I)* X2(K1,I )+YB21(I)* Y2(K1,I )
-      -XB12(I)* X1(K ,I )+YB12(I)* Y1(K ,I )
-      -XB22(I)* X1(K1,I )+YB22(I)* Y1(K1,I )
Y2(K,I) = YA11(I)* D2(K ,I1)+YA21(I)* D2(K1,I1)
-      +YA12(I)* D1(K ,I1)+YA22(I)* D1(K1,I1)
-      -YB21(I)* X2(K1,I )-XB21(I)* Y2(K1,I )
-      -YB12(I)* X1(K ,I )-XB12(I)* Y1(K ,I )
-      -YB22(I)* X1(K1,I )-XB22(I)* Y1(K1,I )
D2(K,I) = XA11(I)* X2(K ,I )+YA11(I)* Y2(K ,I )
-      +XA21(I)* X2(K1,I )+YA21(I)* Y2(K1,I )
-      +XA12(I)* X1(K ,I )+YA12(I)* Y1(K ,I )
-      +XA22(I)* X1(K1,I )+YA22(I)* Y1(K1,I )
-      -XB21(I)* D2(K1,I )-XB12(I)* D1(K ,I )
-      -XB22(I)* D1(K1,I )
600 CONTINUE
C
D(K,L)=D2(K,IEND)
606 CONTINUE
C
DO 660 K=2,KEND
D1(K,1)=D2(K,1)
DO 660 I=2,IEND
D1(K,I)=D2(K,I)
X1(K,I)=X2(K,I)
660 Y1(K,I)=Y2(K,I)
666 CONTINUE
C
CALL ROTAT (D,NX,NY,2,KEND,1)
700 CONTINUE
C
ITIME=ITIME+1
CALL UTTIME (TIME(1,ITIME))
C
C   DETERMINE THE MAXIMUM VALUE
AMXAMP=-0.5E 75

```

```

DO 750 L=2,LEND
DO 750 K=2,KEND
IF(D(K,L).GT.AMXAMP) AMXAMP=D(K,L)
750 CONTINUE
C
C      NORMALIZE
PMAXMA=1./AMXAMP
DO 800 L=2,LEND
DO 800 K=2,KEND
800 D(K,L)=D(K,L)*PMAXMA
C
C      PRINT
IPAGE=0
DO 900 L=J1,LEND,10
LL=L+9
IF(LL.GT.LEND) GO TO 910
IPAGE=IPAGE+1
WRITE(6,6003) IPAGE
900 WRITE(6,6001) ((D(I,J),J=L,LL),I=I1,KEND)
910 WRITE(6,6002) AMXAMP
WRITE(6,6000)
ITIME=ITIME+1
CALL UTTIME (TIME(1,ITIME))
C
C      SET UP WORK AREA FOR PLOTTING
IF(IN.NE.0) GO TO 10
CALL PLTSET (0,'INDIA INK AND SIZE 3 PEN-NIB PLEASE',35)
CALL PLOTST('JOSEP M. COSTA',14,'FEE0505')
IN=1
10 CONTINUE
CALL PLOT(X,Y,-3)
CALL PLOT (0.0 ,0.0 ,2)
CALL PLOT (0.0 ,YSIZE ,3)
CALL PLOT (0.0 ,YSIZE ,2)
CALL PLOT (XSIZE ,YSIZE ,3)
CALL PLOT (XSIZE ,YSIZE ,2)
CALL PLOT (XSIZE ,0.0 ,3)
CALL PLOT (XSIZE ,0.0 ,2)
CALL PLOT (XSIZE2,YSIZE2,3)
CALL PLOT (XSIZE2,YSIZE2,2)
CALL CONTUR (D,NX,NY,GET,NU,I1,KEND,J1,LEND,IPLOT,IPAG,
-XSIZE,YSIZE,NDEC)
ITIME=ITIME+1
CALL UTTIME (TIME(1,ITIME))
X=11.0
Y=0.0
GO TO 1
C
5000 FORMAT (I2,2I4,I2)
5001 FORMAT (8G10.0)

```

```
5002 FORMAT (10G8.0)
6000 FORMAT ('1')
6001 FORMAT ('0',200(10(F12.7,1X)/'0'))
6002 FORMAT('0',G15.7)
6003 FORMAT('1'/'0','PAGE',I3/'0')
6010 FORMAT('1',60(6I20,/,'0'))
END
```

```

C***** MAIN PROGRAM *****
C
C * THIS PROGRAM CONVERTS A ONE-DIMENSIONAL CONTINUOUS FILTER INTO A
C * CIRCULAR TWO-DIMENSIONAL RECURSIVE DIGITAL FILTER WITH A GIVEN
C * CUT-OFF FREQUENCY. IT CALLS THE SUBROUTINE STABIL TO CHECK
C * FOR STABILITY, CALLS THE SUBROUTINE FRES6 TO EVALUATE THE
C * FREQUENCY RESPONSE, AND CALLS THE SUBROUTINE CONTUR TO DRAW A
C * CONTOUR MAP OF THE FREQUENCY RESPONSE.
C
C EXTERNAL FRES1
C INTEGER TITLE1(15), BLANK// 1/
C REAL MAGRES(01,01), PHASER(01,01),
C - GET(09)/.1,.2,.3,.4,.5,.6,.7,.8,.9/
C COMPLEX CMPLX, ZERO1( 10), POLE1( 10), ZERO, POLE, A11(120),
C - A12(120), A21(120),A22(120), B11(120), B12(120), B21(120),
C - B22(120), C0/(0.0,0.0)/, C1/(1.0,0.0)/, CSHIFT, B
C DATA NU/9/
C DATA DEG/.5729577E 02/, PI/.3141593E 01/, RAD/.1745329E-01/
C CALL ERRSET (208,256,-1)
C
C * SET THE VALUES OF THE CONSTANTS
C
C IF MODE=1 THE POLYGONAL HAS A VERTEX ON EACH AXIS
C MODE=2
C MODE=1
C
C IF IPHASE=0 THE TWO-DIMENSIONAL DIGITAL FILTER HAS ZERO-PHASE
C RESPONSE BUT THE NUMBER OF STAGES OF THE FILTER IS DUPLICATED
C IPHASE=0
C IPHASE=1
C
C SHIFT IS AN STABILIZATION FACTOR
C SHIFT=0.0
C
C THETA IS THE DIRECTION IN WHICH THE TWO-DIMENSIONAL FILTER WILL
C HAVE THE DESIRED CUT-OFF FREQUENCY
C THETA=0.0
C
C IF IWORK<1 THE FREQUENCY RESPONSE IS NOT EVALUATED
C IF IWORK<0 STABILITY IS NOT CHECKED
C IWORK=-1
C IWORK=0
C
C XLI=0.0
C XRI=0.999
C IEND=100
C NITMAX=100
C EPS=0.000001
C DFMAX=0.000001
C TENDB=.3162278

```

```
TTYDB=.1
FTYDB=.01
IF(IWORK.LE.0) GO TO 11
C
C * INITIALIZE FOR THE FREQUENCY RESPONSE
NDL=51
IPRINT=1
IPHRES=0
FSCALE=1.0
FR1ADD=0.0
FR2ADD=0.0
C
C * INITIALIZE FOR PLOTTING
CALL PLTSET (0,'INDIA INK PLEASE',16)
CALL PLOTST('JOSEP COSTA',11,'FEE0505')
CALL PLOT(2.0,3.2,-3)
IPAG=1
XSIZE=6.0
YSIZE=6.0
XSIZE2=XSIZE*0.5
YSIZE2=YSIZE*0.5
IPLOT=0
NDEC=10
C
C * READ DATA FILTER CHARACTERISTICS AND PRINT TITLE
11 READ(5,5001,END=2) NROT1,NROT2
   1 READ(5,5000,END=2) CUTOFF, TITLE1
      IF(CUTOFF.NE.0.0) GO TO 3
C
C * END OF EXECUTION
2 IF(IWORK.LE.0) STOP
   CALL PLOTND
   STOP
C
C * READ IN AND PRINT THE POLES AND ZEROS OF THE CONTINUOUS FILTER
3 READ(5,5001) M,N
MAXMN=MAX0(M,N)
DO 30 II=1,MAXMN
POLE1(II)=C0
30 ZERO1(II)=C0
WRITE(6,1001) TITLE1
IF(M.EQ.0) GO TO 50
READ(5,5002) (ZERO1(I),I=1,M)
WRITE(6,6002) (I,ZERO1(I),I=1,M)
WRITE(6,6004)
50 READ(5,5002) (POLE1(I),I=1,N)
WRITE(6,6003) (I,POLE1(I),I=1,N)
NROT11=(NROT1/4)**4
DO 900 NROT=NROT11,NROT2,4
WRITE(6,1000) TITLE1, CUTOFF, NROT
```

```

C
C * DETERMINE THE NUMBER OF FILTERS AND THE ANGLES OF ROTATION
NF=MAXMN*NROT
JEND=NROT
DBETA=360.0/FLOAT(NROT)
BETA0=0.0
IF(MODE.EQ.1) BETA0=DBETA/2.0
IF(IPHASE.EQ.0) GO TO 53
NF=NF/2
JEND=NROT/2
BETA0=BETA0+180.0
C
C * INITIALIZE THE ITERATION
53 WRITE(6,6004)
AA=ANGLE (THETA)
DDD=0.7071068
ZZ=ZEROCCR(DDD)
NIT=0
FRD=CUTOFF
C
C * ITERATION BEGINS
55 DELTA2=PI*FRD/2.0
DO 600 J=1,JEND
BETA=BETA0+FLOAT(J-1)*DBETA
SB=SIN(BETA*RAD)
CB=COS(BETA*RAD)
70 DO 600 I=1,MAXMN
K=(I-1)*JEND+J
IF(NIT.EQ.0) WRITE(6,6010) K,I,BETA
C
C * CALCULATE THE COEFFICIENTS OF THE FILTER
ZERO=CMPLX(DELTA2,0.0)*ZERO1(I)
POLE=CMPLX(DELTA2,0.0)*POLE1(I)
IF(I.GT.M) GO TO 110
A11(K)=CMPLX( CB-SB,0.0)-ZERO
A12(K)=CMPLX(-CB-SB,0.0)-ZERO
A21(K)=CMPLX( CB+SB,0.0)-ZERO
A22(K)=CMPLX(-CB+SB,0.0)-ZERO
GO TO 115
110 A11(K)=C1
A12(K)=A11(K)
A21(K)=A12(K)
A22(K)=A21(K)
115 IF(I.GT.N) GO TO 117
B11(K)=CMPLX( CB-SB,0.0)-POLE
B12(K)=CMPLX(-CB-SB,0.0)-POLE
B21(K)=CMPLX( CB+SB,0.0)-POLE
B22(K)=CMPLX(-CB+SB,0.0)-POLE
GO TO 140
117 B11(K)=C1

```

```

B12(K)=B11(K)
B21(K)=B12(K)
B22(K)=B21(K)
140 IF(ABS(SB).GT.0.01745) GO TO 145
A21(K)=C0
B21(K)=A21(K)
GO TO 150
145 IF(ABS(CB).GT.0.01745) GO TO 400
A12(K)=C0
B12(K)=A12(K)
150 A22(K)=C0
B22(K)=A22(K)
400 IF SHIFT.EQ.0.0 GO TO 500
CSHIFT=CMPLX SHIFT,0.0
B11(K)=B11(K)+CSHIFT*B12(K)
B21(K)=B21(K)+CSHIFT*B22(K)
A21(K)=A21(K)+CSHIFT*A22(K)
A11(K)=A11(K)+CSHIFT*A12(K)

C
C      NORMALIZE WITH RESPECT TO B11(K)
500 B=B11(K)
IF REAL(B).EQ.1.0 .AND. AIMAG(B).EQ.0.01 GO TO 600
A11(K)=A11(K)/B
A12(K)=A12(K)/B
A21(K)=A21(K)/B
A22(K)=A22(K)/B
B11(K)=B11(K)/B
B12(K)=B12(K)/B
B21(K)=B21(K)/B
B22(K)=B22(K)/B
600 CONTINUE
IF(NIT.EQ.0) WRITE(6,6011)
NIT=NIT+1

C      * CHECK THE CUTOFF FREQUENCY
FF=FRESIN (A11,A21,A12,A22,B11,B21,B12,B22,NF)
CALL RTMI (FRC,FRES,FRES1,XLI,XRI,EPS,IEND,IER)
IF(IER.NE.0) WRITE(6,6012) IER
DF=CUTOFF-FRC
FRD=FRD+CUTOFF-FRC
WRITE(6,6015) NIT,CUTOFF,FRC,DF,FRD,FRES
IF(ABS(DF).GE.ABS(DFMAX) .AND. NIT.LT.NITMAX) GO TO 55

C      * END OF ITERATION
IF(NIT.GE.NITMAX) WRITE(6,6012) NITMAX
ZZ=ZEROCCR(0.0)
AA=ANGLE(0.0)

C      * DETERMINE AND PRINT THE SHAPE FACTORS
S1=SHAPE1(0.0,0.05)

```

```

S2=SHAPE1(0.0,0.10)
S3=SHAPE1(45.0,0.05)
S4=SHAPE1(45.0,0.10)
WRITE(6,6201) S1,S2,S3,S4
S1=SHAPE2(0.0,TENDB)
S2=SHAPE2(0.0,TTYDB)
S3=SHAPE2(0.0,FTYDB)
S4=SHAPE2(45.0,TENDB)
S5=SHAPE2(45.0,TTYDB)
S6=SHAPE2(45.0,FTYDB)
WRITE(6,6202) S1,S2,S3,S4,S5,S6
S3=SHAPE3(45)
S4=SHAPE4(45,50)
WRITE(6,6234) S3,S4

C
C * PRINT AND PUNCH THE COEFFICIENTS OF THE FILTER
WRITE(6,6100)
WRITE(6,6110)
WRITE(6,6111) (K,A11(K),A21(K),A12(K),A22(K), K=1,NF)
WRITE(6,6100)
WRITE(6,6120)
WRITE(6,6111) (K,B11(K),B21(K),B12(K),B22(K), K=1,NF)
WRITE(7,5003) TITLE1, CUTOFF, NROT
WRITE(7,5601) (A11(K),A21(K),A12(K),A22(K),K=1,NF)
WRITE(7,5600)
WRITE(7,5601) (B11(K),B21(K),B12(K),B22(K),K=1,NF)
IF(IWORK.LE.0) GO TO 700

C
C * CHECK FOR STABILITY
ISTAB=0
CALL STABIL (B11,B21,B12,B22,NF,ISTAB)
700 IF(IWORK.LE.0) GO TO 900

C
C * EVALUATE THE FREQUENCY RESPONSE
CALL FRES6 (A11,A21,A12,A22,B11,B21,B12,B22,NF,MAGRES,
-           PHASER,NDL,IPRINT,IPHRES,FSCALE,FR1ADD,FR2ADD)

C
C * SET UP WORK AREA FOR PLOTTING
CALL PLOT (0.0    ,0.0    ,2)
CALL PLOT (0.0    ,YSIZE ,3)
CALL PLOT (0.0    ,YSIZE ,2)
CALL PLOT (XSIZE ,YSIZE ,3)
CALL PLOT (XSIZE ,YSIZE ,2)
CALL PLOT (XSIZE ,0.0    ,3)
CALL PLOT (XSIZE ,0.0    ,2)
CALL PLOT (XSIZE2,YSIZE2,3)
CALL PLOT (XSIZE2,YSIZE2,2)

C
C * PLOT A CONTOUR MAP OF THE MAGNITUDE RESPONSE
CALL CONTUR (MAGRES,NDL,NDL,GET,NU,01,NDL,01,NDL,IPLOT,IPAG,

```

```

      -           XSIZE,YSIZE,NDEC)
      CALL PLOT(11.0,0.0,-3)
900 CONTINUE
C
      GO TO 1
C
1000 FORMAT('1',/, '0',15A4,5X,'CUTOFF=',F5.3,5X,'NROT=',I2,/, ' ', 
      -          132('_'),/, '0')
1001 FORMAT('1',15A4,/-)
5000 FORMAT(G6.0,15A4,A2)
5001 FORMAT (3I2)
5002 FORMAT (2G40.0)
5003 FORMAT(15A4,'CUTOFF=',F5.3,' ',' ','NROT=',I2)
5600 FORMAT(2X,78('*'))
5601 FORMAT(8F10.7)
6002 FORMAT ('0','ZERO (',I2,',') = (',G16.7,',',',G16.7,',')')
6003 FORMAT ('0','POLE (',I2,',') = (',G16.7,',',',G16.7,',')')
6004 FORMAT('0','TWO-DIM FILTER #',10X,'ONE-DIM FILTER #',10X,'ROTATION
      - (DEGREES)')
6010 FORMAT('0',13X,I3,23X,I3,17X,F10.3)
6011 FORMAT('1','NIT',10X,'CUTOFF',19X,'FRC',18X,'DF',16X,'FRD',
      -          16X,'FRES',/, '0')
6012 FORMAT(' ','*** ERROR *** VALUE OF IER OR NIT IS ',I3)
6015 FORMAT(' ',I3,5(5X,G16.7),/, '0')
6100 FORMAT ('1// ','COMPLEX COEFFICIENTS OF THE CASCADE OF ROTATED FI
      -LTERS.')
6110 FORMAT ('0','A) COEFFICIENTS OF THE NUMERATOR:','/0',
      -1X,'#',2X,'|',14X,'A11',14X,'|',14X,'A21',14X,'|',14X,'A12',14X,
      -'|',14X,'A22','/+',132('_')/|',4X,4(|',31X))
6111 FORMAT (100(' ',I2,2X,4(|',G15.7,1X,G15.7)/|',4X,4(|',31X)/))
6120 FORMAT ('0','B) COEFFICIENTS OF THE DENOMINATOR:','/0',
      -1X,'#',2X,'|',14X,'B11',14X,'|',14X,'B21',14X,'|',14X,'B12',14X,
      -'|',14X,'B22','/+',132('_')/|',4X,4(|',31X))
6201 FORMAT('1// ','SHAPE1=',4(G16.7, 4X))
6202 FORMAT('0','SHAPE2=',6(G16.7,4X))
6234 FORMAT('0','SHAPE3=',G16.7,'0','SHAPE4=',G16.7)
END

```

SUBROUTINE ROTAT (A,NX,NY,M1,M2,IC)

C THIS SUBROUTINE ROTATES AN SQUARE SUBMATRIX OF A MATRIX BY 0, 90,
C 180, OR 270 DEGREES ACCORDING TO THE VALUE OF IC. THE ROTATION IS
C CLOCKWISE WITH RESPECT TO THE STORING POSITION OF THE MATRIX IN
C THE COMPUTER. THE PRINCIPAL DIAGONAL OF THE SQUARE SUBMATRIX IS
C DEFINED BY A(M1,M1) AND A(M2,M2).

C

C IF IC=0 ROTATE BY 0 DEGREES
C IF IC=1 ROTATE BY 90 DEGREES
C IF IC=2 ROTATE BY 180 DEGREES
C IF IC=3 ROTATE BY 270 DEGREES

C

DIMENSION A(NX,NY)
IROT=IABS(MOD(IC,4))
M12=M1+M2
MED=(M12-1)/2
DO 600 I=M1,MED
IND=M12-I
JND=IND-1
DO 600 J=I,JND
NJ=M12-J
AT=A(I,J)
GO TO (90,180,270),IROT
RETURN
90 A(I,J)=A(NJ,I)
A(NJ,I)=A(IND,NJ)
A(IND,NJ)=A(J,IND)
A(J,IND)=AT
GO TO 600
180 A(I,J)=A(IND,NJ)
A(IND,NJ)=AT
AT=A(J,IND)
A(J,IND)=A(NJ,I)
A(NJ,I)=AT
GO TO 600
270 A(I,J)=A(J,IND)
A(J,IND)=A(IND,NJ)
A(IND,NJ)=A(NJ,I)
A(NJ,I)=AT
600 CONTINUE
RETURN
END

C
C SUBROUTINE CONTUR

JMC 1973 0424 1537

C
C THIS SUBROUTINE IS A MODIFICATION OF THE SUBROUTINE CNTOUR OF
C THE WATSLIB LIBRARY, USEABLE WITH THE UNIVERSITY OF TORONTO
C COMPUTER CENTRE CALCOMP PLOTTING SYSTEM.

C
C PURPOSE

C CONTOURING BY INTERPOLATION OF DATA IN A MATRIX. GIVEN A MATRIX
C CONTAINING DATA, THIS PROGRAMME WILL DRAW A CONTOUR BY
C INTERPOLATING THE APPROXIMATE LOCATION FROM THE LOCATION OF
C THE DATA IN THE MATRIX. A GRID OF FOUR POINTS IS USED TO
C APPROXIMATE A LOCATION OR LOCATIONS IF THE CONTOUR EXISTS IN THE
C REGION. THE PROGRAMME CONTINUES THE SEARCH UNTIL EITHER THE
C LIMITS OF THE MATRIX OR THE LIMITS OF THE AREA WITHIN THE MATRIX
C CONSTRAINED BY THE USER ARE REACHED.

C
C HOW TO USE

C THE USER MUST PREPARE A MAINLINE PROGRAMME WRITTEN ACCORDING TO
C FORTRAN IV SPECIFICATIONS WHICH CONTAINS AT LEAST THE FOLLOWING:

C
C (1) DIMENSION GET(NU), A(NX,NY)

C GET - AN ARRAY OF LENGTH NU CONTAINING THE CONTOUR
C VALUES.
C A - A MATRIX NX BY NY CONTAINING THE DATA.

C
C (2) CALL PLOTST(NAME,NCHAR,CODE)

C NAME - STRING OF UP TO 16 ALPHANUMERIC CHARACTERS
C SPECIFYING THE USER'S NAME TO APPEAR ON THE
C PLOTTED OUTPUT.
C NCHAR - NUMBER OF CHARACTERS IN NAME. NO MORE THAN 16
C CHARACTERS WILL BE PLOTTED.
C CODE - STRING OF 7 CHARACTERS SPECIFYING THE USER'S
C AUTHORIZATION CODE (LETTERS FIRST).

C EXAMPLE: CALL PLOTST ('JOSEP M. COSTA',14,'CAL1973')

C
C (3) CALL CONTUR (A,NX,NY,GET,NU,I1,I2,J1,J2,IPLOT,IPAG,XSIZE,
C YSIZE,NDEC)

C I1 - FIRST ROW TO START CONTOURING IN THE MATRIX.
C I2 - LAST ROW TO FINISH CONTOURING IN THE MATRIX.
C J1 - FIRST COLUMN TO START CONTOURING IN THE MATRIX.
C J2 - LAST COLUMN TO FINISH CONTOURING IN THE MATRIX.
C IPLOT=0 - A LINE IS DRAWN BETWEEN EACH PAIR OF CONSECUTIVE
C POINTS.
C =1 - A + IS DRAWN AT EACH POINT.
C =2 - A X IS DRAWN AT EACH POINT.
C =3 - A * IS DRAWN AT EACH POINT.
C =4 - A O IS DRAWN AT EACH POINT.
C =5 - A | IS DRAWN AT EACH POINT.
C =6 - A Y IS DRAWN AT EACH POINT.

C =7 - NO PLOT.
C >7 - A SYMBOL SPECIFIED BY IPLOT AND THE INTEGER CODE
C OF THE IBM 360 SYMBOL ROUTINE IS DRAWN AT EACH
C POINT.
C <0 - A LINE IS DRAWN BETWEEN EACH PAIR OF CONSECUTIVE
C POINTS AND THE SYMBOL CORRESPONDING TO |IPLOT|
C IS DRAWN AT EACH POINT.
C IPAG=1 - NO PRINTED OUTPUT.
C =2 - PRINTED OUTPUT.
C <0 - THE CONTOUR VALUE IS DRAWN ON CONTOUR AND ONE OF
C THE ABOVE CONDITIONS STILL OCCURS.
C XSIZEx - DIMENSIONS OF THE PLOTTING AREA IN INCHES.
C YSIZEy - IF BOTH XSIZEx AND YSIZEy ARE POSITIVE THESE
C DIMENSIONS CORRESPOND TO THE AREA OCCUPIED BY
C THE MATRIX A.
C - IF EITHER XSIZEx OR YSIZEy IS NEGATIVE THESE
C DIMENSIONS CORRESPOND TO THE AREA CONSTRAINED BY
C THE USER AND LIMITED BY (I1,J1) AND (I2,J2).
C NDEC - CONTROLS THE PRECISION OF THE CONTOUR VALUES FOR
C LABELLING. IF |NDEC|>9 THE CONTOUR IS NOT
C LABELLED.
C >0 - NUMBER OF DIGITS TO THE RIGHT OF THE DECIMAL
C POINTS IN THE CONTOUR VALUES THAT ARE TO BE
C PLOTTED WHEN LABELLING, AFTER PROPER ROUNDING.
C =0 - ONLY THE INTEGER PORTION OF THE CONTOUR VALUES
C AND A DECIMAL POINT ARE PLOTTED, AFTER ROUNDING.
C =-1 - ONLY THE INTEGER PORTION OF THE CONTOUR VALUES
C IS PLOTTED, AFTER ROUNDING.
C <-1 - |NDEC|-1 DIGITS ARE TRUNCATED FROM THE INTEGER
C PORTION AFTER ROUNDING.

(4) CALL PLOTND

C THIS CALL CLOSES THE PLOTTER OUTPUT DATA SET AND PRODUCES
C CLOSING LABELS CONTAINING INFORMATION ABOUT THE AMOUNT OF
C DATA WRITTEN AND THE APPROXIMATE TIME REQUIRED FOR PLOTTING

C REMARKS

C PROGRAMMES CALLING CONTUR USE THE CALCOMP PLOTTER SYSTEM UNLESS
C IPLOT=7 AND CALLS TO THE PLOTST AND PLOTND ROUTINES ARE REMOVED
C FROM THE MAINLINE PROGRAMME.

C REFERENCES

C -SUBROUTINE CNTOUR, WATSLIB LIBRARY.
C -CALCOMP, SECTION 3.7500, USERBOOK, UTCC, TORONTO.

C
C SUBROUTINE CONTUR (A,NX,NY,GET,NU,I1,I2,J1,J2,
C IPLOT,IPAG,XSIZE,YSIZE,NDEC)
C DIMENSION A(NX,NY),GET(NU),X(6),Y(6),MODES(7),NUM(40),CHAR(3),
C XIN(6),YIN(6)

```

EQUIVALENCE (X,XIN), (Y,YIN)
DATA MODES/3,4,11,1,13,9,0/,DEG/.5729577E 02/
C
C   INITIALIZE FOR LABELLING
ISUM=1
INUM=0
ICHECK=0
IF(IPAG.GT.0) GO TO 3
INUM=1
IF(IABS(NDEC).GT.9) INUM=0
IF(INUM.EQ.0) GO TO 3
DO 1 IA=1,NU
1 NUM(IA)=0
C
C   INITIALIZE FOR PRINTING AND PLOTTING
3 IPAGE=IABS(IPAG)
NPLOT=IABS(IPLOT)
IF(NPLOT.EQ.7)IPAGE=2
IF(IPAGE.EQ.2) PRINT 1000
IF(NPLOT.EQ.0.OR.NPLOT.EQ.7) GO TO 5
MODE=NPLOT
IF(MODE.GT.7) MODE=7
MODES(7)=NPLOT
C
C   DETERMINATION OF THE PLOTTING SCALES
5 XS=ABS(XSIZE)
YS=ABS(YSIZE)
IF(XSIZE.LE.0.0 .OR. YSIZE.LE.0.0) GO TO 8
XSCALE=FLOAT(NX-1)/XS
YSCALE=FLOAT(NY-1)/YS
XADD=0.0
YADD=0.0
GO TO 10
8 XSCALE=FLOAT(I2-I1)/XS
YSCALE=FLOAT(J2-J1)/YS
XADD=FLOAT(-I1+1)/XSCALE
YADD=FLOAT(-J1+1)/YSCALE
10 I1P1=I1+1
J2M1=J2-1
C
C   TESTING BY ROWS
DO 600 I=I1P1,I2
IF(INUM.EQ.1) ICHECK=ICHECK+1
FI=I-1
FI1=FI-1
C
C   TESTING BY COLUMNS
DO 600 J=J1,J2M1
FJ=J-1
FJ1=FJ-1

```

```

AIJ=A(I,J)
AI1J=A(I-1,J)
AIJ1=A(I,J+1)
AI1J1=A(I-1,J+1)

C
C      TESTING THE CONTOUR LEVELS
DO 600 NN=1,NU
CL=GET(NN)

C
C      TESTING ONE ELEMENT OF THE GRID
N=0
IS=0
IF(AI1J.LT.CL.AND.AI1J1.GE.CL)GOTO25
IF(AI1J.GE.CL.AND.AI1J1.LT.CL)GOTO25
GOTO30
25 N=N+1
X(N)=FI1
Y(N)=(AI1J-CL)/(AI1J-AI1J1)+FJ
IS=IS+1
30 IF(AI1J.LT.CL.AND.AIJ.GE.CL)GOTO35
IF(AI1J.GE.CL.AND.AIJ.LT.CL)GOTO35
GOTO40
35 N=N+1
Y(N)=FJ
X(N)=(AI1J-CL)/(AI1J-AIJ)+FI1
IS=IS+2
40 IF(AIJ.LT.CL.AND.AIJ1.GE.CL)GOTO45
IF(AIJ.GE.CL.AND.AIJ1.LT.CL)GOTO45
GOTO50
45 N=N+1
X(N)=FI
Y(N)=(AIJ-CL)/(AIJ-AIJ1)+FJ
50 IF(AI1J1.LT.CL.AND.AIJ1.GE.CL)GOTO55
IF(AI1J1.GE.CL.AND.AIJ1.LT.CL)GOTO55
GOTO60
55 N=N+1
Y(N)=FJ1
X(N)=(AI1J1-CL)/(AI1J1-AIJ1)+FI1
IS=IS+4
60 DC=(AI1J+AIJ+AI1J1+AIJ1)*.25
IF(N-2)600,65,120

C
C      DETERMINE THE LOCATION OF THE CONTOUR IN THE GRID (3 POINTS)
65 LX=3
GO TO (90,85,70,80,75,100), IS
70 W=FI1
DX=1.0
Z=FJ
DY=1.0
AL=AI1J-DC

```

```

GOTO105
75 W=FI1
DX=1.0
Z=FJ1
DY=-1.0
AL=AIIJ1-DC
GOTO105
80 Z=FJ1
DY=-1.0
W=FI
DX=-1.0
AL=AIIJ1-DC
GOTO105
85 W=FI
DX=-1.0
Z=FJ
DY=1.0
AL=AIIJ-DC
GOTO105
90 IF(AIIJ.EQ.CL)GOT085
IF(AIJ1.EQ.CL)GOT075
IF(AIIJ1.EQ.CL)GOT080
IF(AIJ.EQ.CL)GOT070
95 X(3)=X(2)
Y(3)=Y(2)
X(2)=0.5*(X(1)+X(3))
Y(2)=0.5*(Y(1)+Y(3))
GO TO 155
100 IF(AIIJ.EQ.CL)GOT075
IF(AIJ1.EQ.CL)GOT085
IF(AIIJ1.EQ.CL)GOT070
IF(AIJ.EQ.CL)GOT080
GOT095
105 V=0.5*(AL+DC-CL)/AL
Y(3)=Y(2)
X(3)=X(2)
X(2)=V*DX+W
Y(2)=V*DY+Z
GO TO 155
C
C      DETERMINE THE LOCATIONS OF THE CONTOUR IN THE GRID (6 POINTS)
120 LX=6
IF(AIIJ-CL)125,130,145
125 IF(DC-CL)135,130,130
130 V=0.5*(AI1J-CL)/(AI1J-DC)
X(6)=X(4)
Y(6)=Y(4)
X(4)=X(3)
Y(4)=Y(3)
X(3)=X(2)

```

```

Y(3)=Y(2)
X(2)=V+FI1
Y(2)=V+FJ
V=-0.5*(AIJ1-CL)/(AIJ1-DC)
X(5)=V+FI
Y(5)=V+FJ1
GO TO 155
135 IF(AIJ-CL)600,150,150
140 IF(AI1J1-CL)130,130,600
145 IF(DC-CL)140,150,150
150 V=0.5*(AI1J1-CL)/(AI1J1-DC)
    X(6)=X(2)
    Y(6)=Y(2)
    XT=X(3)
    YT=Y(3)
    X(3)=X(4)
    Y(3)=Y(4)
    X(4)=XT
    Y(4)=YT
    X(2)=V+FI1
    Y(2)=FJ1-V
    V=-0.5*(AIJ-CL)/(AIJ-DC)
    X(5)=V+FI
    Y(5)=FJ-V
C
C      OUTPUT
155 IPMD=3
DO 165 IA=1,LX
IF(IPAGE.NE.2) GO TO 159
IF(IA.EQ.3) WRITE(6,1001) CL,(X(IY),Y(IY),IY=1,3)
IF(IA.EQ.6) WRITE(6,1001) CL,(X(IY),Y(IY),IY=4,6)
159 XIN(IA)=X(IA)/XSCALE+XADD
YIN(IA)=Y(IA)/YSCALE+YADD
IF(IPLOT.GT.0) GO TO 161
IF(IA.EQ.4) IPMD=3
CALL PLOT (XIN(IA),YIN(IA),IPMD)
IF(IPLOT.EQ.0) GO TO 165
161 IF(NPLOT.EQ.7) GO TO 165
CALL SYMBOL (XIN(IA),YIN(IA),0.07,MODE$1(MODE),0.0,-1)
165 IPMD=2
IF(LX.EQ.6) GO TO 600
C
C      LABELLING THE CONTOUR
370 IF(INUM.NE.1) GO TO 600
IF(ICHECK.LE.0 .OR. NUM(NN).EQ.1) GO TO 600
XINL=XIN(1)
IF(XIN(2).LT.XIN(1))XINL=XIN(2)
YINL=YIN(1)
IF(YIN(2).LT.YIN(1))YINL=YIN(2)
AN=ATAN2(YIN(1)-YIN(2),XIN(1)-XIN(2))

```

```
IF(3.141599-ABS(AN).LT.0.0001) AN=0.0
AN=AN*DEG
IF(ABS(AN).GT.90.0) AN=AN+180.0
XINL=XINL+0.02
YINL=YINL+0.02
ISIGN=1
IF(CL.LT.0) ISIGN=2
NCINT=INT ALOG10(ABS(CL))+1
IF(NCINT.LT.1) NCINT=0
LL=NCINT+NDEC+ISIGN
HEIGHT=0.07
XCOORD=XINL+FLOAT(LL)/XSCALE*COS(AN)*HEIGHT+HEIGHT
YCOORD=YINL+FLOAT(LL)/YSCALE*SIN(AN)*HEIGHT+HEIGHT
IF(XCOORD.GT.XS .OR. XCOORD.LT.0.0 .OR.
- YCOORD.GT.YS .OR. YCOORD.LT.0.0 ) GO TO 600
IF(IPAGE.EQ.2) WRITE(6,1002) CL, AN
NUM(NN)=1
ISUM=ISUM+1
IF(ISUM.GT.NU) INUM=0
IF(NPLOT.EQ.7) GO TO 500
CALL NUMBER (XINL,YINL,HEIGHT,CL,AN,NDEC)
500 ICHECK=0
600 CONTINUE
RETURN
1000 FORMAT('1',1X,'CONTOUR VALUE',15X,3(1HX,12X,1HY,12X)//)
1001 FORMAT(1X,G12.5,10X,6(1X,G12.5))
1002 FORMAT (' ','>>> CONTOUR LABEL ',G12.5,' WILL BE DRAWN HERE AT AN
-ANGLE OF ',F8.3)
END
```

```

FUNCTION FRESIN (A11,A21,A12,A22,B11,B21,B12,B22,NF)
C
C * THIS FUNCTION CALCULATES THE MAGNITUDE RESPONSE AT A POINT
C * GIVEN IN CARTESIAN COORDINATES OR IN POLAR COORDINATES
C
C * FRESIN SUPPLIES THE FILTER COEFFICIENTS AND DETERMINES
C * THE NORMALIZATION FACTOR FOR A PEAK RESPONSE OF 1.0
C
LOGICAL IN
COMPLEX CMPLX,CEXP,A11(NF),A21(NF),A12(NF),A22(NF),
-           B11(NF),B21(NF),B12(NF),B22(NF),
-           EE12,EE21,EE22,F2
DATA PI/.3141593E 01/, RAD/.1745329E-01/
IN=.FALSE.
EE21=(1.0,0.0)
EE12=EE21
EE22=EE12
DELTA=0.0
UNE=1.0
GO TO 5
C
C * CARTESIAN COORDINATES
ENTRY FRESO (FR1,FR2)
PIFR1=PI*FR1
PIFR2=PI*FR2
GO TO 4
C
C * POLAR COORDINATES
ENTRY ANGLE(THETA)
PICT=PI*COS(THETA*RAD)
PIST=PI*SIN(THETA*RAD)
ANGLE=THETA
RETURN
ENTRY FRES1 (RFR)
PIFR1=PICT*RFR
PIFR2=PIST*RFR
4 EE21=CEXP(CMPLX(0.0,PIFR1))
EE12=CEXP(CMPLX(0.0,PIFR2))
EE22=CEXP(CMPLX(0.0,PIFR1+PIFR2))
5 F2=(1.0,0.0)
DO 6 K=1,NF
6 F2=F2*(A11(K)+A21(K)*EE21+A12(K)*EE12+A22(K)*EE22)/
-           (B11(K)+B21(K)*EE21+B12(K)*EE12+B22(K)*EE22)
C1   -   /(10.0,0.0)
C
C * IN CASE OF OVERFLOW MAKE EFFECTIVE THE SCALING
C * OPERATION INDICATED IN COMMENT C1 ABOVE
C
FRESO=(CABS(F2)*UNE)-DELTA
FRES1=FRESO

```

```
IF(IN) RETURN
IN=.TRUE.
UNE=1.0/FRES1
FRESIN=UNE
C
C * ENTRY ZEROCR IS USED IN CONJUNCTION WITH SUBROUTINE RTMI
ENTRY ZEROCR (DDD)
DELTA=DDD
ZEROCR=0.0
RETURN
END
```

SUBROUTINE RTMI

REFERENCE:

IBM APPLICATION PROGRAM
SCIENTIFIC SUBROUTINE PACKAGE
(360A-CM-03X) VERSION III
APPLICATION DESCRIPTION
SIXTH EDITION (IBM FORM NO. H20-0166-5)
PAGES 217-219.

```

FUNCTION SHAPE1 (THETA,DELTA)
C
C THIS SUBPROGRAM CALCULATES FOUR DIFFERENT SHAPE FACTORS OF THE
C MAGNITUDE RESPONSE OF A TWO-DIMENSIONAL DIGITAL FILTER.
C QUADRANTAL SYMMETRY OF THE MAGNITUDE RESPONSE OF THE FILTER
C IS ASSUMED.
C *** N.B. BEFORE CALLING ANY OF THE FUNCTIONS IN THIS SUBPROGRAM
C           THE COEFFICIENTS OF THE TWO-DIMENSIONAL DIGITAL FILTER
C           MUST BE SUPPLIED BY CALLING THE FUNCTION FRESIN.
C
C EXTERNAL FRES1
DATA CUTOFF/0.1/, ICUT/10/, AA1/.7071068/, XLI/0.0/, XRI/.9999999/,
-   EPS/.000001/, IEND/100/, PI/.3141593E 01/
AA=ANGLE(THETA)
ZZ=ZEROCCR(AA1)
CALL RTMI (FRC,FRES,FRES1,XLI,XRI,EPS,IEND,IER)
IF(IER.NE.0) WRITE(6,6000) IER
ZZ=ZEROCCR(0.0)
FR=FRC+DELTA
AA2=FRES1(FR)
SHAPE1=20.0 ALOG10(AA1/AA2)
RETURN
C
C ENTRY SHAPE2 (THETA,BB2)
AA=ANGLE(THETA)
ZZ=ZEROCCR(AA1)
CALL RTMI (FRC,FRES,FRES1,XLI,XRI,EPS,IEND,IER)
IF(IER.NE.0) WRITE(6,6000) IER
ZZ=ZEROCCR(BB2)
CALL RTMI (FR,FRES,FRES1,XLI,XRI,EPS,IEND,IER)
IF(IER.NE.0) WRITE(6,6000) IER
SHAPE2=FR-FRC
RETURN
C
C ENTRY SHAPE3 (INUM)
ZZ=ZEROCCR(AA1)
THETAD=PI/FLOAT(INUM+INUM)
SHAPE3=0.0
DO 3 I=1,INUM
THETAI=THETAD*(FLOAT(I)-0.5)
AA=ANGLE(THETAI)
CALL RTMI (FRC,FRES,FRES1,XLI,XRI,EPS,IEND,IER)
IF(IER.NE.0) WRITE(6,6000) IER
V2=CUTOFF+ABS(CUTOFF-FRC)
3 SHAPE3=V2*V2+SHAPE3
SHAPE3=(SHAPE3/FLOAT(INUM)-CUTOFF*CUTOFF)*PI
RETURN
C

```

```
C
C      ENTRY SHAPE4 (INUM, IDIV)
C
C *** N.B.: IDIV SHOULD BE DIVISIBLE BY 10.
C
C      DDD=0.0
C      ZZ=ZEROCCR(DDD)
C      THETAD=PI/FLOAT(INUM+INUM)
C      SHAPE4=0.0
C      JCUT=IDIV/ICUT
C      DO 6 I=1, INUM
C          THETA1=THETAD*(FLOAT(I)-0.5)
C          AA=ANGLE(THETA1)
C          DO 5 J=1, JCUT
C              5 SHAPE4=(1.0-FRES1((FLOAT(J)-0.5)/FLOAT(IDIV)))*FLOAT(J+J-1)+SHAPE4
C              JCUT1=JCUT+1
C              DO 6 J=JCUT1, IDIV
C                  6 SHAPE4=FRES1((FLOAT(J)-0.5)/FLOAT(IDIV))*FLOAT(J+J-1)+SHAPE4
C                  SHAPE4=SHAPE4*PI/FLOAT(IDIV*IDIV*INUM)
C
C      RETURN
C      6000 FORMAT('0','> *** ERROR *** > IN SUBROUTINE SHAPE IER HAS THE VALU
C                  -E ',I2)
C      END
```

```

C SUBROUTINE STABIL (B11,B12,B21,B22,NF,ISTAB)
C DETERMINES IF A CASCADE OF BILINEAR FILTERS IS STABLE
C
C MEANING OF THE ARGUMENT ISTAB:
C INPUT:   ISTAB>0 ONLY THE UNSTABLE FILTERS OF THE CASCADE ARE PR
C           ISTAB<0 ONLY THE STABLE FILTERS OF THE CASCADE ARE PR
C           ISTAB=0 ALL THE FILTERS IN THE CASCADE ARE PRINTED
C OUTPUT:  ISTAB=1 THE CASCADE OF BILINEAR FILTERS IS STABLE
C           ISTAB=0 AT LEAST ONE FILTER IN THE CASCADE IS UNSTABLE
C
COMPLEX CMPLX, B11(NF), B12(NF), B21(NF), B22(NF), CENTRE,
-     CINF//INFINITY//
COMPLEX*16 FMT(6)/*(1H0,11X,2A4,14X*,*,I2,15X*,*
-                   *1H*,G15.7,1H,,G1*,*5.7,1H),1X,G15.7*,*
-                   *,1X,G15.7,1X,*,*G15.7)*/, FMT3, FMT4, FMT6,
-                   CENINF/*12X,2A4,17X,*/, RADINF/*A4,4HNITY,4X*/,
-                   STAINF/*4X,A4,4HNITY)*/
DATA AINF//INFI//,STA// STA//,BLE//BLE //,UNST//UNST//,
-     ABLE//ABLE//,QUESTM//? ? ? //
FMT3=FMT(3)
FMT4=FMT(4)
FMT6=FMT(6)
ISINT=ISTAB
ISTAB=1
R1=0.99996
R2=1.0
IINF=0
JINDEX=1
WRITE(6,6000)
DO 600 K=1,NF
BM11=CABS(B11(K))
BM12=CABS(B12(K))
BM21=CABS(B21(K))
BM22=CABS(B22(K))
IF(BM11*BM22.NE.BM12*BM21) GO TO 100
IF(BM22.NE.0.0) GO TO 80
IF(BM12.NE.0.0) GO TO 65
IF(ABS(BM11/BM21).LT.1.0) GO TO 75
GO TO 70
65 IF(BM21.NE.0.0) GO TO 80
IF(ABS(BM11/BM12).GT.1.0) GO TO 75
70 IF(ISINT.GT.0) GO TO 150
S1=STA
S2=BLE
GO TO 90
75 ISTAB=0
IF(ISINT.GT.0) GO TO 150
S1=UNST

```

```

S2=ABLE
GO TO 90
80 S1=QUESTM
S2=QUESTM
90 WRITE(6,9001) S1, S2, K
JINDEX=1
GO TO 600
100 IF(BM12.NE.BM22) GO TO 105
CENTRE=CINF
RADIUS=AINF
IINF=1
FMT(3)=CENINF
FMT(4)=RADINF
STAB1=0.0
GO TO 125
105 IF(BM12.NE.0.0) GO TO 110
CENTRE=B21(K)/B22(K)
GO TO 120
110 IF(BM22.NE.0.0) GO TO 115
CENTRE=B11(K)/B12(K)
GO TO 120
115 BM12=BM12*BM12
BM22=BM22*BM22
CENTRE=(B12(K)*B21(K)*CMPLX(BM22,0.0)-
- B11(K)*B22(K)*CMPLX(BM12,0.0))/(
- (B12(K)*B22(K)*CMPLX(BM12-BM22,0.0)))
120 RADIUS=CABS(B11(K)*B22(K)-B21(K)*B12(K))/ABS(BM12-BM22)
STAB1=CABS(CENTRE)-RADIUS
125 IF(BM21.NE.0.0) GO TO 130
IINF=1
FMT(6)=STAINF
STAB2=AINF
IF(ABS(STAB1).GT.R1) GO TO 140
GO TO 137
130 STAB2=CABS(B11(K)/B21(K))
135 IF(ABS(STAB1).GT.R1 .AND. STAB2.GT.R2) GO TO 140
137 ISTAB=0
IF(ISINT.GT.0) GO TO 150
S1=UNST
S2=ABLE
GO TO 145
140 IF(ISINT.LT.0) GO TO 150
S1=STA
S2=BLE
145 WRITE(6,FMT) S1, S2, K, CENTRE, RADIUS, STAB1, STAB2
JINDEX=1
IF(IINF.EQ.0) GO TO 600
147 IINF=0
FMT(3)=FMT3
FMT(4)=FMT4

```

```
FMT(6)=FMT6
GO TO 600
150 IF(JINDEX.EQ.1) WRITE(6,1002)
JINDEX=0
IF(IINF.EQ.1) GO TO 147
600 CONTINUE
RETURN
1002 FORMAT (* *,30X,**)
6000 FORMAT (*1/* *,10X,'STABILITY',11X,'FILTER #',23X,'C E N T R E',
-           16X,'RADIUS',11X,'STAB1',11X,'STAB2'/*+',10X,9(*_),11X,
-           8(*_),23X,11(*_),16X,6(*_),11X,5(*_),11X,5(*_*)/*',
-           50X,'(REAL PART , IMAGINARY PART)/*/*')
9001 FORMAT (*0*,11X,2A4,14X,I2,07X,'MESSAGE: SEE COEFFICIENTS OF THIS
-FILTER.')
END
```

```

SUBROUTINE FRES6 (A11,A12,A21,A22,B11,B12,B21,B22,NF,MAGRES,
-                  PHASER,NDL,IPRINT,IPHASE,FSCALE,FR1ADD,FR2ADD)

C
C THIS SUBROUTINE EVALUATES THE FREQUENCY RESPONSE (MAGNITUDE AND/OR
C PHASE) OF A TWO-DIMENSIONAL RECURSIVE FILTER GIVEN AS A CASCADE OF
C BILINEAR FILTERS WITH COMPLEX COEFFICIENTS.
C

REAL MAGRES(NDL,NDL), PHASER(NDL,NDL), MAXMAG, FR1(51), FR2(51)
COMPLEX CMPLX,CEXP,A11(NF),A12(NF),A21(NF),A22(NF),B11(NF),B12(NF)
- ,B21(NF),B22(NF),EE12,EE21,EE22,F2
DATA PI/.3141593E 01/
NDL2=(NDL/2)+1
NDL1=NDL2+NDL2
NDL=NDL1-1
DNDL=FLOAT(NDL-1)
MAXMAG=0.0
DO 6 N=1,NDL
FR2(N)=FSCALE*FLOAT(N+N-NDL1)/DNDL+FR2ADD
PIFR2=PI*FR2(N)
DO 6 M=1,NDL
FR1(M)=FSCALE*FLOAT(M+M-NDL1)/DNDL+FR1ADD
PIFR1=PI*FR1(M)
EE12=CEXP(CMPLX(0.0,PIFR2))
EE21=CEXP(CMPLX(0.0,PIFR1))
EE22=CEXP(CMPLX(0.0,PIFR1+PIFR2))
F2=(1.0,0.0)
DO 4 K=1,NF
4 F2=(A11(K)+A12(K)*EE12+A21(K)*EE21+A22(K)*EE22)/
- (B11(K)+B12(K)*EE12+B21(K)*EE21+B22(K)*EE22)*F2
IF(IPHASE.EQ.1) GO TO 3
MAGRES(M,N)=CABS(F2)
GO TO 5
3 XMAG=REAL(F2)
YMAG=AIMAG(F2)
MAGRES(M,N)=SQRT(XMAG*XMAG+YMAG*YMAG)
IF(XMAG.NE.0.0 .OR. YMAG.NE.0.0) GO TO 7
PHASER(M,N)=0.0
GO TO 5
7 CONTINUE
PHASER(M,N)=ATAN2(YMAG,XMAG)

C
C DETERMINATION OF THE PEAK RESPONSE
5 IF(MAGRES(M,N).GT.MAXMAG) MAXMAG=MAGRES(M,N)
6 CONTINUE

C
C NORMALIZATION TO A PEAK RESPONSE OF 1.0
GAMXAM=1.0/MAXMAG
DO 8 N=1,NDL
DO 8 M=1,NDL
8 MAGRES(M,N)=MAGRES(M,N)*GAMXAM

```

```
9 IF(IPRINT.EQ.0) GO TO 9999
C
C      PRINT THE FREQUENCY RESPONSE
NN00=NDL2
NN00=NDL-20
N20=NN00+20
WRITE(6,1001) (FR2(N),N=NN00,N20), (FR1(M), (MAGRES(M,N),
-          N=NN00,N20), M=1,NDL)
WRITE(6,1002) MAXMAG
IFI(IPRINT.LE.0 .OR. IPHASE.NE.1) GO TO 9999
WRITE(6,1003) (FR2(N),N=NN00,N20), (FR1(M), (PHASER(M,N),
-          N=NN00,N20), M=1,NDL)
1001 FORMAT ('1',57X,'MAGNITUDE RESPONSE'/'+',57X,18(''_')//
-          '0', 'FR2=' ,2X,21(F5.2,1X)/'0', 'FR1'/'+' , '___=' /' /
-          '0',255(F5.2,'|',21(F5.3,1X)/' '))
1002 FORMAT ('0', 'MAXMAG=' ,G15.8)
1003 FORMAT ('1',59X,'PHASE RESPONSE'/'+',59X,14(''_')//
-          '0', 'FR2=' ,2X,21(F5.2,1X)/'0', 'FR1='/'+' , '___' /' /
-          '0',255(F5.2,'|',21F6.3/' '))
9999 RETURN
END
```

REFERENCES

- [1] 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.
- [2] E.L. Hall, "A Comparison of Computations for Spatial Frequency Filtering", Proc. of the IEEE, vol. 60, No. 7, pp. 887-891, July 1972.
- [3] 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. 115-128, June 1972.
- [4] W.H. Lyne and J.P. Mantey, "Computer Implementation of Two-Dimensional Recursive Digital Filters", in SWIEECO Record, p. 19F1, April 1969.
- [5] O.J. Tretiak and T.S. Huang, "Separable-Sum Approximation of Two-Dimensional Digital Filters", in Proc. UMR-Mervin J. Kelly Comm. Conf., Missouri, October 1970, pp. 10-3-1 to 10-3-3.
- [6] J.L. Shanks, "Two Planar Digital Filter Algorithms", in Proc. of the 5th Princeton Conf. on Information Sciences & Systems, pp. 48-53, March 1971.
- [7] S. Treitel and J.L. Shanks, "The Design of Multistage Separable Planar Filters", IEEE Trans. on Geoscience Electronics, vol. GE-9, No. 1, pp. 10-27, January 1971.
- [8] T.S. Huang, "Two-Dimensional Windows", IEEE Trans. on Audio and Electroacoustics (Correspondence), Vol. AU-20, pp. 88-89, March 1972.
- [9] J. Hu, "Frequency Sampling Design of Two-Dimensional Finite Impulse Response Digital Filters", M.S. Thesis, Mass. Inst. Tech., Cambridge, June 1972.

- [10] C.H. Farmer and D.S. Gooden, "Rotation and Stability of a Recursive Digital Filter", in Proc. of the Two-Dimensional Digital Signal Processing Conference, Columbia, Missouri, pp. 1-2-1 to 1-2-12, October 1971.
- [11] L.R. Rabiner and B. Gold, "Processing of Two-Dimensional Signals", unpublished notes, May 1972.
- [12] R.M. Selzer, "Improving Biomedical Image Quality with Computers", JPL Technical Report 32-1336, October 1968.
- [13] O.J. Tretiak, "The Rationale of Picture Filtering", in Proc. of the Two-Dimensional Digital Signal Processing Conference, Columbia, Missouri, pp. 5-3-1 to 5-3-1, October 1971.
- [14] A. Rosenfeld, Picture Processing by Computer, New York: Academic Press, 1969.
- [15] 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.
- [16] J.B. Knowles and R. Edwards, "Complex Cascade Programming and Associated Computational Errors", Electronic Letters, Vol. 1, No. 6, pp. 160-161, August 1965.
- [17] W. Rudin, Real and Complex Analysis, McGraw-Hill, New York: 1966, p. 270.
- [18] J.L. Shanks and R.R. Read, "Two-Dimensional Recursive Filters for Digital Processing", in Proc. of the 1973 International Symposium on Circuit Theory, Toronto, Ontario, April 1973, pp. 32-35.

- [19] J.B. Knowles and R. Edwards, "Effect of a finite-word-length computer in a sampled-data feedback system", Proc. IEE, vol. 112, no. 6, pp. 1197-1207, June 1965.
- [20] M. Corrington, "Design of Two-Dimensional Recursive Filters", presented at the 1973 International Symposium on Circuit Theory, Toronto, Ontario, April 1973.
- [21] J.H. Justice and J.L. Shanks, "Stability Criterion for N-Dimensional Digital Filters", IEEE Trans. on Automatic Control, vol. AC-18, no. 3, pp. 284-286, June 1973.
- [22] R.R. Read and S. Treitel, "The Stabilization of Two-Dimensional Recursive Filters Via the Discrete Hilbert Transform", IEEE Trans. on Geoscience Electronics, vol. GE-11, no. 3, pp. 153-160, July 1973.
- [23] B.D.O. Anderson and E.I. Jury, "Stability Test for Two-Dimensional Recursive Filters", IEEE Trans. on Audio and Electroacoustics, vol. AU-21, no. 4, pp. 366-372, August 1973.

TABLE I
COEFFICIENTS OF THE FILTER SHOWN IN FIGURE 1

RE(A11)	IM(A11)	RE(A21)	IM(A21)	RE(A12)	IM(A12)	RE(A22)	IM(A22)	BETA
0.7920657	0.1439704	0.0	0.0	0.7920657	0.1439704	0.0	0.0	0.0
0.7920657	-0.1439704	0.0	0.0	0.7920657	-0.1439704	0.0	0.0	0.0
RE(B11)	IM(B11)	RE(B21)	IM(B21)	RE(B12)	IM(B12)	RE(B22)	IM(B22)	BETA
1.0000000	0.0	0.0	0.0	0.5841307	-0.2879409	0.0	0.0	0.0
1.0000000	0.0	0.0	0.0	-0.5841307	0.2879409	0.0	0.0	0.0

TABLE II
COEFFICIENTS OF THE FILTER SHOWN IN FIGURE 2

RE(A11)	IM(A11)	RE(A21)	IM(A21)	RE(A12)	IM(A12)	RE(A22)	IM(A22)	BETA
0.6752226	0.1036685	0.6752226	0.1036685	0.6752226	0.1036685	0.6752226	0.1036685	285.00
0.6752226	-0.1036685	0.6752226	-0.1036685	0.6752226	-0.1036685	0.6752226	-0.1036685	285.00
RE(B11)	IM(B11)	RE(B21)	IM(B21)	RE(B12)	IM(B12)	RE(B22)	IM(B22)	BETA
1.0000000	0.0	-0.3044304	-0.2002723	0.6504826	-0.0536623	-0.6539479	-0.2539346	285.00
1.0000000	0.0	-0.3044304	0.2002723	0.6504826	0.0536623	-0.6539479	0.2539346	285.00

TABLE III
COEFFICIENTS OF THE FILTER SHOWN IN FIGURE 3

RE(A11)	IM(A11)	RE(A21)	IM(A21)	RE(A12)	IM(A12)	RE(A22)	IM(A22)	RETA
0.6000551	0.0814605	0.6000551	0.0814605	0.6000551	0.0814605	0.6000551	0.0814605	315.00
0.6000551	-0.0814605	0.6000551	-0.0814605	0.6000551	-0.0814605	0.6000551	-0.0814605	315.00
RE(B11)	IM(B11)	RE(B21)	IM(B21)	RE(B12)	IM(B12)	RE(B22)	IM(B22)	RETB
1.0000000	0.0	0.1513925	-0.1152030	0.1513968	-0.1152024	-0.6972111	-0.2304053	315.00
1.0000000	0.0	0.1513925	0.1152030	0.1513968	0.1152024	-0.6972111	0.2304053	315.00

TABLE IV
COEFFICIENTS OF THE FILTER SHOWN IN FIGURE 4

RE(A11)	IM(A11)	RE(A21)	IM(A21)	RE(A12)	IM(A12)	RE(A22)	IM(A22)	BETA
0.6752208	0.1036680	0.6752208	0.1036680	0.6752208	0.1036680	0.6752208	0.1036680	345.00
0.	-0.1036680	0.	-0.1036680	0.	-0.1036680	0.	-0.1036680	345.00
RE(B11)	IM(B11)	RE(B21)	IM(B21)	RE(B12)	IM(B12)	RE(B22)	IM(B22)	BETA
1.0000000	0.0	0.6504766	-0.0536631	-0.3044255	-0.2002711	-0.6539489	-0.2539341	345.00
1.0000000	0.0	0.6504766	0.0536631	-0.3044255	0.2002711	-0.6539489	0.2539341	345.00

TABLE V

STABILITY REGIONS FOR ROTATED FILTERS

Transfer function of the two-dimensional filter	Location of the poles of the original one-dimensional filter $H(s)$	
	L.H.P.	R.H.P.
$H(z_1, z_2)$	$270^\circ < \beta < 360^\circ$	$90^\circ < \beta < 180^\circ$
$H(z_1^{-1}, z_2)$	$0^\circ < \beta < 90^\circ$	$180^\circ < \beta < 270^\circ$
$H(z_1, z_2^{-1})$	$180^\circ < \beta < 270^\circ$	$0^\circ < \beta < 90^\circ$
$H(z_1^{-1}, z_2^{-1})$	$90^\circ < \beta < 180^\circ$	$270^\circ < \beta < 360^\circ$

COEFFICIENTS OF THE FILTER SHOWN IN FIGURE 8

TABLE VI

RE(A11)	IM(A11)	RE(A21)	IM(A21)	RE(A12)	IM(A12)	RE(A22)	IM(A22)	BETA
0.6752226	0.1036685	0.6752226	0.1036685	0.6752226	0.1036685	0.6752226	0.1036685	285.00
0.6000551	0.0814605	0.6000551	0.0814605	0.6000551	0.0814605	0.6000551	0.0814605	315.00
0.6752208	0.1036680	0.6752208	0.1036680	0.6752208	0.1036680	0.6752208	0.1036680	345.00
0.6752226	-0.1036685	0.6752226	-0.1036685	0.6752226	-0.1036685	0.6752226	-0.1036685	285.00
0.6000551	-0.0814605	0.6000551	-0.0814605	0.6000551	-0.0814605	0.6000551	-0.0814605	315.00
0.6752208	-0.1036680	0.6752208	-0.1036680	0.6752208	-0.1036680	0.6752208	-0.1036680	345.00
RE(B11)	IM(B11)	RE(B21)	IM(B21)	RE(B12)	IM(B12)	RE(B22)	IM(B22)	BETA
1.0000000	0.0	-0.3044304	-0.2002723	0.6504826	-0.0536623	-0.6539479	-0.2539346	285.00
1.0000000	0.0	0.1513925	-0.1152030	0.1513968	-0.1152024	-0.6972111	-0.2304053	315.00
1.0000000	0.0	0.6504766	-0.0536631	-0.3044255	-0.2002711	-0.6539489	-0.2539341	345.00
1.0000000	0.0	-0.3044304	0.2002723	0.6504826	0.0536623	-0.6539479	0.2539346	285.00
1.0000000	0.0	0.1513925	0.1152030	0.1513968	0.1152024	-0.6972111	0.2304053	315.00
1.0000000	0.0	0.6504766	0.0536631	-0.3044255	0.2002711	-0.6539489	0.2539341	345.00

TABLE VII
COEFFICIENTS OF THE FILTER SHOWN IN FIGURE 9

RE(A11)	IM(A11)	RE(A21)	IM(A21)	RE(A12)	IM(A12)	RE(A22)	IM(A22)	BETA
-1.7043905	0.7807194	-1.7043905	0.7807184	-1.7043905	0.7807184	-1.7043905	0.7807184	195.00*
2.2507906	2.2508221	2.2507906	2.2508221	2.2507906	2.2508221	2.2507906	2.2508221	225.00*
1.0179634	0.2433522	1.0179634	0.2433522	1.0179634	0.2433522	1.0179634	0.2433522	255.00*
-1.7043905	-0.7807184	-1.7043905	-0.7807184	-1.7043905	-0.7807184	-1.7043905	-0.7807184	195.00*
2.2507906	-2.2508221	2.2507906	-2.2508221	2.2507906	-2.2508221	2.2507906	-2.2508221	225.00*
1.0179634	-0.2433522	1.0179634	-0.2433522	1.0179634	-0.2433522	1.0179634	-0.2433522	255.00*
RE(B11)	IM(B11)	RE(B21)	IM(B21)	RE(B12)	IM(B12)	RE(B22)	IM(B22)	BETA
1.00300000	0.0	1.8822527	-0.4041275	-2.2926302	1.5082312	-1.4103765	1.1041040	195.00*
1.00000000	0.0	-2.1830912	-3.1831360	4.1831055	3.1831493	1.0000134	0.0000137	225.00*
1.00000000	0.0	-0.9665530	-0.4701202	1.5269394	0.1259691	-0.4396124	-0.3441508	255.00*
1.00000000	0.0	1.8822527	0.4041275	-2.2926302	-1.5082312	-1.4103765	-1.1041040	195.00*
1.00000000	0.0	-2.1830912	3.1831360	4.1831055	-3.1831493	1.0000134	-0.0000137	225.00*
1.00000000	0.0	-0.9665530	0.4701202	1.5269394	-0.1259691	-0.4396124	0.3441508	255.00*

TABLE VIII
COEFFICIENTS OF THE FILTER SHOWN IN FIGURE 10

RE(A11)	IM(A11)	RE(A21)	IM(A21)	RE(A12)	IM(A12)	RE(A22)	IM(A22)	BETA
2.2507906	2.2508221	2.2507906	2.2508221	2.2507906	2.2508221	2.2507906	2.2508221	225.00*
0.6000551	0.7814605	0.6000551	0.0814605	0.6000551	0.0814605	0.6000551	0.0814605	315.00
2.2507906	-2.2508221	2.2507906	-2.2508221	2.2507906	-2.2508221	2.2507906	-2.2508221	225.00*
0.6000551	-0.0814605	0.6000551	-0.0814605	0.6000551	-0.0814605	0.6000551	-0.0814605	315.00

RF(B11)	IM(B11)	RE(B21)	IM(B21)	RE(B12)	IM(B12)	RE(B22)	IM(B22)	BETA
1.0000000	0.0	-2.1830912	-3.1831360	4.1831055	3.1831493	1.0000134	0.0000137	225.00*
1.0000000	0.0	0.1513925	-0.1152030	0.1513968	-0.1152024	-0.6972111	-0.2304053	315.00
1.0000000	0.0	-2.1830912	3.1831360	4.1831055	-3.1831493	1.0000134	-0.0000137	225.00*
1.0000000	0.0	0.1513925	0.1152030	0.1513968	0.1152024	-0.6972111	0.2304053	315.00

TABLE IX
COEFFICIENTS OF THE FILTER SHOWN IN FIGURE 11

RE(A11)	IM(A11)	RE(A21)	IM(A21)	RE(A12)	IM(A12)	RE(A22)	IM(A22)	BETA
-2.1109352	1.4697571	-2.1109352	1.4697571	-2.1109352	1.4697571	-2.1109352	1.4697571	202.50*
1.2077503	0.3514758	1.2077503	0.3514758	1.2077503	0.3514758	1.2077503	0.3514758	247.50*
0.6406206	0.0930918	0.6406206	0.0930918	0.6406206	0.0930918	0.6406206	0.0930918	292.50
0.6406198	0.3930916	0.6406198	0.0930916	0.6406198	0.0930916	0.6406198	0.0930916	337.50
-2.1109352	-1.4697571	-2.1109352	-1.4697571	-2.1109352	-1.4697571	-2.1109352	-1.4697571	202.50*
1.2077503	-0.3514758	1.2077503	-0.3514758	1.2077503	-0.3514758	1.2077503	-0.3514758	247.50*
0.6406206	-0.0930918	0.6406206	-0.0930918	0.6406206	-0.0930918	0.6406206	-0.0930918	292.50
0.6406198	-0.3930916	0.6406198	-0.0930916	0.6406198	-0.0930916	0.6406198	-0.0930916	337.50
RE(B11)	IM(B11)	RE(B21)	IM(B21)	RE(B12)	IM(B12)	RE(B22)	IM(B22)	BETA
1.0000000	0.0	2.6156321	-1.1248980	-2.9005013	2.7157583	-1.2848673	1.5908585	202.50*
1.0000000	0.0	-1.2316303	-0.6494422	1.9243755	0.2690089	-0.3804331	-0.3804331	247.50*
1.0000000	0.0	-0.1837130	-0.1720114	0.5096928	-0.0712491	-0.6740203	-0.2432605	292.50
1.0000000	0.0	0.5096887	-0.0712496	-0.1837092	-0.1720107	-0.6740206	-0.2432603	337.50
1.0000000	0.0	2.6156321	1.1248980	-2.9005013	-2.7157583	-1.2848673	-1.5908585	202.50*
1.0000000	0.0	-1.2316303	0.6494422	1.9243755	0.2690089	-0.3804331	-0.3804331	247.50*
1.0000000	0.0	-0.1837130	0.1720114	0.5096928	0.0712491	-0.6740203	-0.2432605	292.50
1.0000000	0.0	0.5096887	-0.0712496	-0.1837092	-0.1720107	-0.6740206	-0.2432603	337.50

TABLE X

COEFFICIENTS OF THE FILTER SHOWN IN FIGURE 12

RE(A11)	IM(A11)	RE(A21)	IM(A21)	RE(A12)	IM(A12)	RE(A22)	IM(A22)	BETA
-1.7043905	0.7807184	-1.7043905	0.7807184	-1.7043905	0.7807184	-1.7043905	0.7807184	195.00*
2.2507936	2.2508221	2.2507906	2.2508221	2.2507906	2.2508221	2.2507906	2.2508221	225.00*
1.0179634	0.2433522	1.0179634	0.2433522	1.0179634	0.2433522	1.0179634	0.2433522	255.00*
0.6752226	0.1036685	0.6752226	0.1036685	0.6752226	0.1036685	0.6752226	0.1036685	285.00
0.6000551	0.0814605	0.6000551	0.0814605	0.6000551	0.0814605	0.6000551	0.0814605	315.00
0.6752208	0.1036680	0.6752208	0.1036680	0.6752208	0.1036680	0.6752208	0.1036680	345.00
-1.7043905	-0.7807184	-1.7043905	-0.7807184	-1.7043905	-0.7807184	-1.7043905	-0.7807184	195.00*
2.2507906	-2.2508221	2.2507906	-2.2508221	2.2507906	-2.2508221	2.2507906	-2.2508221	225.00*
1.0179634	-0.2433522	1.0179634	-0.2433522	1.0179634	-0.2433522	1.0179634	-0.2433522	255.00*
0.6752226	-0.1036685	0.6752226	-0.1036685	0.6752226	-0.1036685	0.6752226	-0.1036685	285.00
0.6000551	-0.0814605	0.6000551	-0.0814605	0.6000551	-0.0814605	0.6000551	-0.0814605	315.00
0.6752208	-0.1036680	0.6752208	-0.1036680	0.6752208	-0.1036680	0.6752208	-0.1036680	345.00
1.0179634	-0.2433522	1.0179634	-0.2433522	1.0179634	-0.2433522	1.0179634	-0.2433522	225.00*
0.6752226	-0.1036685	0.6752226	-0.1036685	0.6752226	-0.1036685	0.6752226	-0.1036685	255.00*
0.6000551	-0.0814605	0.6000551	-0.0814605	0.6000551	-0.0814605	0.6000551	-0.0814605	315.00
0.6752208	-0.1036680	0.6752208	-0.1036680	0.6752208	-0.1036680	0.6752208	-0.1036680	345.00
RE(B11)	IM(B11)	RE(B21)	IM(B21)	RE(B12)	IM(B12)	RE(B22)	IM(B22)	BETA
1.7030000	0.0	1.8822527	-0.4041275	-2.2926302	1.5082312	-1.4103765	1.1041040	195.00*
0.0000000	0.0	-2.1830912	-3.1831360	4.1831055	3.1831493	1.0000134	0.0000137	225.00*
1.0000000	0.0	-0.9665530	-0.4701202	1.5269394	0.1259691	-0.4396124	-0.3441508	255.00*
1.0000000	0.0	-0.3044304	-0.2002723	0.6504326	-0.0536623	-0.6539479	-0.2539346	285.00
1.0000000	0.0	0.1513925	-0.1152030	0.1513968	-0.1152024	-0.6972111	-0.2304053	315.00
1.0000000	0.0	0.6504766	-0.0536631	0.3044255	-0.2027111	-0.6539489	-0.2539341	345.00
1.0000000	0.0	1.8822527	0.4041275	-2.2926302	-1.5082312	-1.4103765	-1.1041040	195.00*
1.0000000	0.0	-2.1830912	3.1831360	4.1831055	-3.1831493	1.0000134	-0.0000137	225.00*
1.0000000	0.0	-0.9665530	0.4701202	1.5269394	-0.1259691	-0.4396124	-0.3441508	255.00*
1.0000000	0.0	-0.3044304	0.2002723	0.6504326	0.0536623	-0.6539479	-0.2304053	285.00
1.0000000	0.0	0.1513925	0.1152030	0.1513968	0.1152024	-0.6972111	-0.2539341	315.00
1.0000000	0.0	0.6504766	0.0536631	0.3044255	-0.2027111	-0.6539489	-0.2539341	345.00

TABLE XI
COEFFICIENTS OF THE FILTER SHOWN IN FIGURE 13

RE(A11)	IM(A11)	RE(A21)	IM(A21)	RE(A12)	IM(A12)	RE(A22)	IM(A22)	BETA
-1.5358171	0.6053960	-1.5358171	0.6053960	-1.5358171	0.6053960	-1.5358171	0.6053960	191.25*
-1.0290947	4.2525463	-1.0290947	4.2525463	-1.0290947	4.2525463	-1.0290947	4.2525463	213.75*
1.6746902	0.7469719	1.6746902	0.7469719	1.6746902	0.7469719	1.6746902	0.7469719	236.25*
0.9462518	0.2085696	0.9462518	0.2085696	0.9462518	0.2085696	0.9462518	0.2085696	258.75*
0.6976829	0.1108613	0.6976829	0.1108613	0.6976829	0.1108613	0.6976829	0.1108613	281.25
0.6098124	0.0841832	0.6098124	0.0841832	0.6098124	0.0841832	0.6098124	0.0841832	303.75
0.6098120	0.0841830	0.6098120	0.0841830	0.6098120	0.0841830	0.6098120	0.0841830	326.25
0.6976810	0.1108607	0.6976810	0.1108607	0.6976810	0.1108607	0.6976810	0.1108607	348.75
-1.5358171	-0.6053960	-1.5358171	-0.6053960	-1.5358171	-0.6053960	-1.5358171	-0.6053960	191.25*
-1.0290947	-4.2525463	-1.0290947	-4.2525463	-1.0290947	-4.2525463	-1.0290947	-4.2525463	213.75*
1.6746902	-0.7469719	1.6746902	-0.7469719	1.6746902	-0.7469719	1.6746902	-0.7469719	236.25*
0.9462518	-0.2085696	0.9462518	-0.2085696	0.9462518	-0.2085696	0.9462518	-0.2085696	258.75*
0.6976829	-0.1108613	0.6976829	-0.1108613	0.6976829	-0.1108613	0.6976829	-0.1108613	281.25
0.6098124	-0.0841832	0.6098124	-0.0841832	0.6098124	-0.0841832	0.6098124	-0.0841832	303.75
0.6098120	-0.0841830	0.6098120	-0.0841830	0.6098120	-0.0841830	0.6098120	-0.0841830	326.25
0.6976810	-0.1108607	0.6976810	-0.1108607	0.6976810	-0.1108607	0.6976810	-0.1108607	348.75
RE(B11)	IM(B11)	RE(B21)	IM(B21)	RE(B12)	IM(B12)	RE(B22)	IM(B22)	BETA
1.0000000	0.0	1.5992393	-0.2362111	-2.0126143	1.1875267	-1.4133739	0.9513150	191.25*
1.0000000	0.0	2.1434650	-4.7251616	-0.7113241	7.0717344	0.4321416	2.3465710	213.75*
1.0000000	0.0	-1.7849054	-1.2421665	2.8608208	0.8299916	0.0759158	-0.4121747	236.25*
1.0000000	0.0	-0.8561385	-0.4091238	1.3692112	0.0813802	-0.4869268	-0.3277434	258.75*
1.0000000	0.0	-0.3685545	-0.2174624	0.7277807	-0.0432556	-0.6407737	-0.2607178	281.25
1.0000000	0.0	-0.0140817	-0.1399916	0.3224157	-0.0935390	-0.6916666	-0.2335308	303.75
1.0000000	0.0	0.3224120	-0.0935395	-0.0140782	-0.1399912	-0.6916668	-0.2335307	326.25
1.0000000	0.0	0.7277746	-0.0432564	-0.3685493	-0.2174610	-0.6407746	-0.2607173	348.75
1.0000000	0.0	1.5992393	0.2362111	-2.0126143	-1.1875267	-1.4133739	-0.9513150	191.25*
1.0000000	0.0	2.1434650	4.7251616	-0.7113241	-7.0717344	0.4321416	-2.3465710	213.75*
1.0000000	0.0	-1.7849054	-1.2421665	2.8608208	-0.8299916	0.0759158	-0.4121747	236.25*
1.0000000	0.0	-0.8561385	-0.4091238	1.3692112	-0.0913802	-0.4869268	-0.3277434	258.75*
1.0000000	0.0	-0.3685545	-0.2174624	0.7277807	-0.0432556	-0.6407737	-0.2607178	281.25
1.0000000	0.0	-0.0140817	-0.1399916	0.3224157	-0.0935390	-0.6916666	-0.2335308	303.75
1.0000000	0.0	0.3224120	-0.0935395	-0.0140782	-0.1399912	-0.6916668	-0.2335307	326.25
1.0000000	0.0	0.7277746	-0.0432564	-0.3685493	-0.2174610	-0.6407746	-0.2607173	348.75

TABLE XII

COEFFICIENTS OF THE FILTER SHOWN IN FIGURE 14

RE(A11)	IM(A11)	RE(A21)	IM(A21)	RE(A12)	IM(A12)	RE(A22)	IM(A22)	BETA
-1.4490061	0.5284554	-1.4490061	0.5284554	-1.4490061	0.5284554	-1.4490061	0.5284554	189.00*
-2.2495451	2.3256598	-2.2495451	2.3256598	-2.2495451	2.3256598	-2.2495451	2.3256598	207.00*
2.2507906	2.2508221	2.2507906	2.2508221	2.2507906	2.2508221	2.2507906	2.2508221	225.00*
1.3623552	0.4591305	1.3623552	0.4591305	1.3623552	0.4591305	1.3623552	0.4591305	243.00*
0.9088913	0.1916705	0.9088913	0.1916705	0.9088913	0.1916705	0.9088913	0.1916705	261.00*
0.7130716	0.1159399	0.7130716	0.1159399	0.7130716	0.1159399	0.7130716	0.1159399	279.00
0.6255363	0.0886705	0.6255363	0.0886705	0.6255363	0.0886705	0.6255363	0.0886705	297.00
0.6050551	0.0814605	0.6050551	0.0814605	0.6050551	0.0814605	0.6050551	0.0814605	315.00
0.6255355	0.0886703	0.6255355	0.0886703	0.6255355	0.0886703	0.6255355	0.0886703	333.00
0.7130701	0.1159393	0.7130701	0.1159393	0.7130701	0.1159393	0.7130701	0.1159393	351.00
-1.4490061	-0.5284554	-1.4490061	-0.5284554	-1.4490061	-0.5284554	-1.4490061	-0.5284554	189.00*
-2.2495451	-2.3256598	-2.2495451	-2.3256598	-2.2495451	-2.3256598	-2.2495451	-2.3256598	207.00*
2.2507906	2.2508221	2.2507906	2.2508221	2.2507906	2.2508221	2.2507906	2.2508221	225.00*
1.3623552	-0.4591305	1.3623552	-0.4591305	1.3623552	-0.4591305	1.3623552	-0.4591305	243.00*
0.9088913	-0.1916705	0.9088913	-0.1916705	0.9088913	-0.1916705	0.9088913	-0.1916705	261.00*
0.7130716	-0.1159399	0.7130716	-0.1159399	0.7130716	-0.1159399	0.7130716	-0.1159399	279.00
0.6255363	-0.0886705	0.6255363	-0.0886705	0.6255363	-0.0886705	0.6255363	-0.0886705	297.00
0.6000551	-0.0814605	0.6000551	-0.0814605	0.6000551	-0.0814605	0.6000551	-0.0814605	315.00
0.6255355	-0.0886703	0.6255355	-0.0886703	0.6255355	-0.0886703	0.6255355	-0.0886703	333.00
0.7130701	-0.1159393	0.7130701	-0.1159393	0.7130701	-0.1159393	0.7130701	-0.1159393	351.00
RE(B11)	IM(B11)	RE(B21)	IM(B21)	RE(B12)	IM(B12)	RE(B22)	IM(B22)	BETA
1.0000000	0.0	1.4533434	-0.1653354	-1.8623333	1.0438976	-1.4089890	0.8785630	189.00*
1.0000000	0.0	3.0425367	-2.1116467	-3.0087204	4.1443577	-0.9661831	2.0327101	207.00*
1.0000000	0.0	-2.1830912	-3.1831360	4.1831055	3.1831493	1.0000134	0.0000137	225.00*
1.0000000	0.0	-1.4277315	-0.8181756	2.2369976	0.4168833	-0.1907351	-0.4012923	243.00*
1.0000000	0.0	-0.7954009	-0.3786212	1.2843566	0.0599684	-0.5110340	-0.3186528	261.00*
1.0000000	0.0	-0.4085850	-0.2290249	0.7769050	-0.0362736	-0.6316802	-0.2652985	279.00
1.0000000	0.0	-0.1147139	-0.1580122	0.4320276	-0.0805108	-0.6326866	-0.2385230	297.00
1.0000000	0.0	0.1513925	-0.1152030	0.1513968	-0.1152024	-0.6972111	-0.2304053	315.00
1.0000000	0.0	0.4320236	-0.0805113	0.147104	-0.1580114	-0.6826870	-0.2385228	333.00
1.0000000	0.0	0.7768299	-0.0362743	-0.4085808	-0.2290238	-0.6316810	-0.2652981	351.00
1.0000000	0.0	1.4533434	0.1653354	-1.8623333	1.0438976	-1.4089890	-0.8785630	189.00*
1.0000000	0.0	3.0425367	2.1116467	-3.0087204	4.1443577	-0.9661831	-0.327101	207.00*
1.0000000	0.0	-2.1830912	3.1831360	4.1831055	-3.1831493	1.0000134	-0.0000137	225.00*
1.0000000	0.0	-1.4277315	0.8181756	2.2369976	-0.4168833	-0.1907351	-0.4012923	243.00*
1.0000000	0.0	-0.7954009	0.3786212	1.2843566	-0.0599684	-0.5110340	-0.3186528	261.00*
1.0000000	0.0	-0.4085850	0.2290249	0.7769050	-0.0362736	-0.6316802	-0.2652985	279.00
1.0000000	0.0	-0.1147139	0.1580122	0.4320276	-0.0805108	-0.6826866	-0.2385230	297.00
1.0000000	0.0	0.1513925	0.1152030	0.1513968	-0.1152024	-0.6972111	-0.2304053	315.00
1.0000000	0.0	0.4320236	0.0805113	-0.147104	-0.1580114	-0.6826870	-0.2385228	333.00
1.0000000	0.0	0.7768299	-0.0362743	-0.4085808	-0.2290238	-0.6316810	-0.2652981	351.00

TABLE XIII
COEFFICIENTS OF THE FILTER SHOWN IN FIGURE 15

RE(A11)	IM(A11)	RE(A21)	IM(A21)	RE(A12)	IM(A12)	RE(A22)	IM(A22)	BETA
-1.3965874	0.4856843	-1.3965874	0.4856843	-1.3965874	0.4856843	-1.3965874	0.4856843	187.50*
-2.1109352	1.4697571	-2.1109352	1.4697571	-2.1109352	1.4697571	-2.1109352	1.4697571	202.50*
0.7397731	4.3765383	0.7397731	4.3765383	0.7397731	4.3765383	0.7397731	4.3765383	217.50*
1.8937187	1.0242836	1.8937187	1.0342836	1.8937187	1.0342836	1.8937187	1.0342836	232.50*
1.2077503	0.3514758	1.2077503	0.3514758	1.2077503	0.3514758	1.2077503	0.3514758	247.50*
0.8860354	0.1817327	0.8860354	0.1817327	0.8860354	0.1817327	0.8860354	0.1817327	262.50*
0.7242043	0.1196906	0.7242043	0.1196906	0.7242043	0.1196906	0.7242043	0.1196906	277.50
0.6406206	0.0930918	0.6406206	0.0930918	0.6406206	0.0930918	0.6406206	0.0930918	292.50
0.6043614	0.0826563	0.6043614	0.0826563	0.6043614	0.0826563	0.6043614	0.0826563	307.50
0.6043611	0.0826563	0.6043611	0.0826563	0.6043611	0.0826563	0.6043611	0.0826563	322.50
0.6406198	0.0930916	0.6406198	0.0930916	0.6406198	0.0930916	0.6406198	0.0930916	337.50
0.7242017	0.1196898	0.7242017	0.1196898	0.7242017	0.1196898	0.7242017	0.1196898	352.50
-1.3965874	-0.4856843	-1.3965874	-0.4856843	-1.3965874	-0.4856843	-1.3965874	-0.4856843	187.50*
-2.1109352	-1.4697571	-2.1109352	-1.4697571	-2.1109352	-1.4697571	-2.1109352	-1.4697571	202.50*
0.7397731	-4.3765383	0.7397731	-4.3765383	0.7397731	-4.3765383	0.7397731	-4.3765383	217.50*
1.8937187	-1.0342836	1.8937187	-1.0342836	1.8937187	-1.0342836	1.8937187	-1.0342836	232.50*
1.2077503	-0.3514758	1.2077503	-0.3514758	1.2077503	-0.3514758	1.2077503	-0.3514758	247.50*
0.8860354	-0.1817327	0.8860354	-0.1817327	0.8860354	-0.1817327	0.8860354	-0.1817327	262.50*
0.7242043	-0.1196906	0.7242043	-0.1196906	0.7242043	-0.1196906	0.7242043	-0.1196906	277.50
0.6406206	-0.0930918	0.6406206	-0.0930918	0.6406206	-0.0930918	0.6406206	-0.0930918	292.50
0.6043614	-0.0826563	0.6043614	-0.0826563	0.6043614	-0.0826563	0.6043614	-0.0826563	307.50
0.6043611	-0.0826563	0.6043611	-0.0826563	0.6043611	-0.0826563	0.6043611	-0.0826563	322.50
0.6406198	-0.0930916	0.6406198	-0.0930916	0.6406198	-0.0930916	0.6406198	-0.0930916	337.50
0.7242017	-0.1196898	0.7242017	-0.1196898	0.7242017	-0.1196898	0.7242017	-0.1196898	352.50

TABLE XIII
COEFFICIENTS OF THE FILTER SHOWN IN FIGURE 15

RE(B11)	IM(B11)	RE(B21)	IM(B21)	RE(B12)	IM(B12)	RE(B22)	IM(B22)	BETA
1.00000000	0.0	1.3645763	-0.1267869	-1.7692795	0.9630582	-1.4947012	0.8362712	187.50*
1.00000000	0.0	2.6156321	-1.1248980	-2.9005013.	2.7157583	-1.2848673	1.5908585	202.50*
1.00000000	0.0	0.993125	-5.3285189	2.1738033	6.9442911	1.2731161	1.6157713	217.50*
1.00000000	0.0	-2.047731	-1.6411037	3.3056507	1.2592659	0.3008775	-0.3818370	232.50*
1.00000000	0.0	-1.2316303	-0.6494422	1.9243755	0.2690689	-0.3072541	-0.3804331	247.50*
1.00000000	0.0	-0.7569085	-0.3603557	1.2313942	0.0474424	-0.5256044	-0.3129135	262.50*
1.00000000	0.0	-0.4360164	-0.2373333	0.8109488	-0.0312450	-0.6250679	-0.2685782	277.50
1.00000000	0.0	-0.1837130	-0.1726114	0.5096928	-0.0712491	-0.6740203	-0.2432605	292.50
1.00000000	0.0	0.0410548	-0.1311516	0.2641791	-0.1006356	-0.6947666	-0.2317873	307.50
1.00000000	0.0	0.2641746	-0.1006362	0.0410590	-0.1311510	-0.6947668	-0.2317873	322.50
1.00000000	0.0	0.5096887	-0.0712496	-0.1837092	-0.1720107	-0.6740206	-0.2432603	337.50
1.00000000	0.0	0.8109418	-0.0312460	-0.4360109	-0.2373314	-0.6250693	-0.2685774	352.50
1.00000000	0.0	1.3645763	0.1267869	-1.7692795	-0.9630582	-1.4947012	-0.8362712	187.50*
1.00000000	0.0	2.6156321	1.1248980	-2.9005013	-2.7157583	-1.2848673	-1.5908585	202.50*
1.00000000	0.0	0.993125	5.3285189	2.1738033	6.9442911	1.2731161	1.6157713	217.50*
1.00000000	0.0	-2.047731	1.6411037	3.3056507	1.2592659	0.3008775	-0.3818370	232.50*
1.00000000	0.0	-1.2316303	0.6494422	1.9243755	0.2690089	-0.3072541	-0.3804331	247.50*
1.00000000	0.0	-0.7569085	0.3603557	1.2313042	-0.0474424	-0.5256044	-0.3129135	262.50*
1.00000000	0.0	-0.4360164	0.2373333	0.8109488	0.0312450	-0.6250679	-0.2685782	277.50
1.00000000	0.0	-0.1837130	0.1720114	0.5096928	0.0712491	-0.6740203	-0.2432605	292.50
1.00000000	0.0	0.0410548	0.1311516	0.2641791	0.1006356	-0.6947666	-0.2317873	307.50
1.00000000	0.0	0.2641746	0.1006362	0.0410590	0.1311510	-0.6947668	-0.2317873	322.50
1.00000000	0.0	0.5096887	0.0712496	-0.1837092	-0.1720107	-0.6740206	-0.2432603	337.50
1.00000000	0.0	0.8109418	0.0312460	-0.4360109	-0.2373314	-0.6250693	-0.2685774	352.50

TABLE XIV

CPU TIME IN SECONDS EMPLOYED IN OBTAINING
THE DATA FOR FIGS. 23 THROUGH 29

Figure	Solution of the recursive equations	Normalization to a peak value of 1.0 and print output	Determination of the location of the contour levels
23	0.99	0.82	2.74
24	2.84	0.82	2.37
25	2.48	0.82	2.00
26	1.98	1.63	3.85
27	5.00	1.62	2.11
28	3.85	1.80	6.52
29	9.87	1.82	2.68

TABLE XV

SHAPE FACTOR NUMBER 1 (in dB)

Order of Butterworth filter	Number of rotated filters being cascaded					
	2	4	6	8	10	12
$d=0.05 \quad \theta=0^\circ$						
1	2.80	3.05	3.28	3.41	3.50	3.56
2	7.06	7.34	8.56	9.26	9.77	10.15
3	12.59	12.75	15.56	17.54	19.10	20.36
4	18.96	19.04	23.48	27.53	30.80	33.63
5	25.78	25.82	31.75	38.51	43.92	48.86
6	32.83	32.84	40.15	49.98	57.72	65.15
7	39.98	39.98	48.60	61.67	71.82	81.99
8	47.17	47.17	57.10	73.43	86.05	99.07
9	54.38	54.38	65.65	85.20	100.30	116.27
10	61.59	61.60	74.26	96.94	114.53	133.49
$d=0.05 \quad \theta=45^\circ$						
1	1.95	2.99	3.21	3.34	3.32	3.48
2	5.50	7.17	8.34	9.02	9.51	9.87
3	9.71	12.46	15.12	17.05	18.54	19.74
4	14.19	18.61	23.05	26.76	29.91	32.60
5	18.79	25.26	31.62	37.46	42.70	47.41
6	23.42	32.16	40.48	48.66	56.26	63.27
7	28.06	39.18	49.46	60.09	70.20	79.70
8	32.72	46.25	58.44	71.60	84.32	96.39
9	37.37	53.34	67.37	83.13	98.50	113.20
10	42.02	60.44	76.25	94.64	112.70	130.04

TABLE XV

SHAPE FACTOR NUMBER 1 (in dB)

Order of Butterworth filter	Number of rotated filters being cascaded					
	2	4	6	8	10	12
$d=0.1 \quad \theta=0^\circ$						
1	5.75	6.55	7.26	7.70	8.01	8.24
2	15.41	16.74	21.28	24.21	26.53	28.44
3	27.06	28.25	39.43	46.68	53.35	59.28
4	39.33	40.25	58.96	71.18	83.51	94.85
5	51.76	52.44	78.78	96.15	114.61	131.97
6	64.23	64.73	98.56	121.18	145.83	169.56
7	76.71	77.07	118.27	146.16	176.94	207.29
8	89.20	89.46	137.92	171.09	207.88	245.00
9	101.68	101.87	157.52	195.97	238.65	282.65
10	114.16	114.30	177.10	220.80	269.27	320.20
$d=0.1 \quad \theta=45^\circ$						
1	3.74	6.35	7.02	7.45	7.74	7.95
2	10.51	16.20	20.49	23.29	25.48	27.27
3	18.03	27.40	37.33	44.95	51.30	56.90
4	25.69	39.12	55.08	68.66	80.50	91.26
5	33.39	51.06	72.74	92.83	110.13	127.22
6	41.09	63.10	90.28	117.05	141.25	163.68
7	48.80	75.19	107.65	141.22	171.84	200.28
8	56.50	87.33	124.87	165.32	202.41	236.85
9	64.21	99.49	141.97	189.37	232.95	273.45
10	71.92	111.67	158.97	213.37	263.44	309.93

TABLE XVI

SHAPE FACTOR NUMBER 2

Order of Butterworth filter	Number of rotated filters being cascaded					
	2	4	6	8	10	12
$A=10 \quad \theta=0^\circ$						
1	0.1210	0.1060	0.0968	0.0922	0.0895	0.0877
2	0.0496	0.0480	0.0428	0.0406	0.0393	0.0384
3	0.0309	0.0307	0.0279	0.0263	0.0254	0.0247
4	0.0224	0.0224	0.0210	0.0196	0.0189	0.0184
5	0.0176	0.0176	0.0169	0.0157	0.0151	0.0147
6	0.0145	0.0145	0.0141	0.0132	0.0126	0.0122
7	0.0123	0.0123	0.0121	0.0114	0.0109	0.0105
8	0.0107	0.0107	0.0106	0.0101	0.0096	0.0092
9	0.0094	0.0094	0.0094	0.0090	0.0086	0.0083
10	0.0085	0.0085	0.0084	0.0082	0.0078	0.0075
$A=10 \quad \theta=45^\circ$						
1	0.2081	0.1093	0.0996	0.0947	0.0919	0.0900
2	0.0639	0.0489	0.0436	0.0413	0.0400	0.0391
3	0.0361	0.0313	0.0283	0.0267	0.0258	0.0251
4	0.0249	0.0228	0.0211	0.0199	0.0191	0.0186
5	0.0190	0.0178	0.0170	0.0160	0.0153	0.0149
6	0.0153	0.0147	0.0143	0.0134	0.0128	0.0124
7	0.0128	0.0125	0.0125	0.0116	0.0110	0.0107
8	0.0110	0.0108	0.0112	0.0102	0.0097	0.0094
9	0.0096	0.0096	0.0102	0.0091	0.0087	0.0084
10	0.0086	0.0086	0.0094	0.0083	0.0078	0.0076

TABLE XVI

SHAPE FACTOR NUMBER 2

Order of Butterworth filter	Number of rotated filters being cascaded					
	2	4	6	8	10	12
$A=20 \quad \theta=0^\circ$						
1	0.3049	0.2369	0.2066	0.1907	0.1814	0.1753
2	0.1098	0.1013	0.0841	0.0775	0.0735	0.0709
3	0.0647	0.0636	0.0533	0.0489	0.0463	0.0446
4	0.0456	0.0454	0.0398	0.0360	0.0341	0.0327
5	0.0351	0.0351	0.0321	0.0289	0.0271	0.0260
6	0.0285	0.0285	0.0270	0.0241	0.0226	0.0217
7	0.0240	0.0240	0.0232	0.0209	0.0195	0.0186
8	0.0208	0.0208	0.0204	0.0185	0.0171	0.0164
9	0.0183	0.0183	0.0181	0.0166	0.0154	0.0146
10	0.0163	0.0163	0.0162	0.0151	0.0139	0.0132
$A=20 \quad \theta=45^\circ$						
1	0.6874	0.2526	0.2185	0.2006	0.1903	0.1837
2	0.1832	0.1043	0.0864	0.0794	0.0753	0.0726
3	0.0930	0.0650	0.0545	0.0499	0.0472	0.0454
4	0.0608	0.0463	0.0400	0.0367	0.0347	0.0333
5	0.0448	0.0357	0.0318	0.0293	0.0276	0.0264
6	0.0354	0.0290	0.0266	0.0245	0.0230	0.0220
7	0.0292	0.0244	0.0231	0.0212	0.0198	0.0189
8	0.0248	0.0211	0.0206	0.0188	0.0174	0.0166
9	0.0216	0.0185	0.0188	0.0169	0.0156	0.0148
10	0.0191	0.0165	0.0173	0.0153	0.0141	0.0134

TABLE XVI

SHAPE FACTOR NUMBER 2

Order of Butterworth filter	Number of rotated filters being cascaded					
	2	4	6	8	10	12
$A=40 \quad \theta=0^\circ$						
1	0.6532	0.4694	0.3967	0.3545	0.3288	0.3120
2	0.2546	0.2007	0.1565	0.1371	0.1264	0.1194
3	0.1392	0.1283	0.0951	0.0843	0.0775	0.0731
4	0.0937	0.0920	0.0694	0.0614	0.0564	0.0531
5	0.0702	0.0699	0.0558	0.0486	0.0447	0.0420
6	0.0559	0.0559	0.0471	0.0405	0.0372	0.0349
7	0.0465	0.0465	0.0412	0.0349	0.0320	0.0300
8	0.0397	0.0397	0.0366	0.0308	0.0281	0.0263
9	0.0346	0.0346	0.0329	0.0277	0.0252	0.0236
10	0.0307	0.0307	0.0298	0.0253	0.0228	0.0214
$A=40 \quad \theta=45^\circ$						
1	0.8898	0.5498	0.4541	0.3944	0.3622	0.3414
2	0.6127	0.2117	0.1619	0.1422	0.1308	0.1234
3	0.2800	0.1329	0.0992	0.0865	0.0794	0.0748
4	0.1661	0.0945	0.0720	0.0628	0.0575	0.0542
5	0.1149	0.0716	0.0565	0.0496	0.0455	0.0427
6	0.0869	0.0571	0.0466	0.0412	0.0378	0.0355
7	0.0696	0.0473	0.0399	0.0355	0.0325	0.0305
8	0.0578	0.0404	0.0351	0.0313	0.0286	0.0268
9	0.0494	0.0352	0.0315	0.0282	0.0257	0.0239
10	0.0431	0.0312	0.0288	0.0257	0.0233	0.0217

TABLE XVIISHAPE FACTOR NUMBER 3 ($\times 10^{-6}$)

Order of Butterworth filter	Number of rotated filters being cascaded					
	2	4	6	8	10	12
1	4.38	0.19	0.14	0.23	0.21	0.20
2	6.02	0.32	0.14	0.14	0.16	0.16
3	16.14	1.94	0.34	0.16	0.21	0.16
4	27.15	1.09	0.40	0.12	0.32	0.97
5	38.08	0.19	0.67	0.13	0.16	0.21
6	48.90	1.73	1.05	0.40	0.22	0.37
7	59.69	3.72	1.24	0.44	0.16	0.19
8	70.15	5.47	1.02	0.56	0.37	0.84
9	81.00	7.24	1.03	0.95	0.83	0.37
10	90.93	9.25	0.20	1.02	0.78	0.85

TABLE XVIIISHAPE FACTOR NUMBER 4 ($\times 10^{-2}$)

Order of Butterworth filter	Number of rotated filters being cascaded					
	2	4	6	8	10	12
1	17.65	12.09	9.98	8.92	8.31	7.93
2	4.53	3.90	3.16	2.88	2.72	2.62
3	2.34	2.24	1.86	1.71	1.62	1.56
4	1.55	1.53	1.33	1.21	1.15	1.11
5	1.15	1.14	1.03	0.94	0.89	0.86
6	0.90	0.90	0.84	0.76	0.72	0.69
7	0.73	0.73	0.70	0.64	0.60	0.57
8	0.61	0.61	0.59	0.54	0.50	0.48
9	0.51	0.51	0.50	0.46	0.42	0.40
10	0.43	0.43	0.42	0.39	0.36	0.33

FIGURE CAPTIONS

- Fig. 1. Contour plot of the two-dimensional magnitude response of a second-order Butterworth filter rotated 0° .
- Fig. 2. Contour plot of the two-dimensional magnitude response of a second-order Butterworth filter rotated 285° .
- Fig. 3. Contour plot of the two-dimensional magnitude response of a second-order Butterworth filter rotated 315° .
- Fig. 4. Contour plot of the two-dimensional magnitude response of a second-order Butterworth filter rotated 345° .
- Fig. 5. Fine detail of the contour plot in Fig. 3 at high frequencies.
- Fig. 6. Stabilization of marginally-unstable rotated filters.
- Fig. 7. Regions of stability for rotated filters with transfer function $H(z_1, z_2)$.
- Fig. 8. Contour plot of the two-dimensional magnitude response of a cascade of three second-order Butterworth filters rotated 285° , 315° , and 345° .
- Fig. 9. Contour plot of the two-dimensional magnitude response of a cascade of three second-order Butterworth filters rotated 195° , 225° , and 255° .
- Fig. 10. Contour plot of the two-dimensional magnitude response of a cascade of two second-order Butterworth filters rotated 225° and 315° .
- Fig. 11. Contour plot of the two-dimensional magnitude response of a cascade of four second-order Butterworth Filters rotated by multiples of 45° .

- Fig. 12. Contour plot of the two-dimensional magnitude response of a cascade of six second-order Butterworth filters rotated by multiples of 30° .
- Fig. 13. Contour plot of the two-dimensional magnitude response of a cascade of eight second-order Butterworth filters rotated by multiples of 22.5° .
- Fig. 14. Contour plot of the two-dimensional magnitude response of a cascade of ten second-order Butterworth filters rotated by multiples of 18° .
- Fig. 15. Contour plot of the two-dimensional magnitude response of a cascade of twelve second-order Butterworth filters rotated by multiples of 15° .
- Fig. 16. Fine detail of one quadrant of the contour plot in Fig. 10.
- Fig. 17. Fine detail of one quadrant of the contour plot in Fig. 11.
- Fig. 18. Fine detail of one quadrant of the contour plot in Fig. 12.
- Fig. 19. Fine detail of one quadrant of the contour plot in Fig. 13.
- Fig. 20. Fine detail of one quadrant of the contour plot in Fig. 14.
- Fig. 21. Fine detail of one quadrant of the contour plot in Fig. 15.
- Fig. 22. Block diagram for two-dimensional complex cascade programming.
- Fig. 23. Contour map of the impulse response of a second-order Butterworth filter rotated 315° .
- Fig. 24. Contour map of the impulse response of a second-order Butterworth filter rotated 225° .
- Fig. 25. Contour map of the impulse response of the filter shown in Fig. 8.

- Fig. 26. Contour map of the impulse response of the filter shown in Fig. 10.
- Fig. 27. Contour map of the impulse response of the filter shown in Fig. 12.
- Fig. 28. Contour map of the impulse response of a filter with zero-phase response and magnitude response the square of the magnitude response shown in Fig. 10.
- Fig. 29. Contour map of the impulse response of a filter with zero-phase response and magnitude response the square of the magnitude response shown in Fig. 12.
- Fig. 30. Block diagrams for the elements in Fig. 22 including the computational errors.
- Fig. 31. Noise model for two-dimensional complex cascade programming.
- Fig. 32. Magnitude responses, in the direction 0° , of 18 two-dimensional recursive filters with cutoff frequencies from 0.05 to 0.9 at intervals of 0.05. Each filter consists of a cascade of four second-order Butterworth filters rotated by 202.5° , 247.5° , 292.5° , and 337.5° .
- Fig. A.1. Mapping $z_2 = f(z_1)$.
- Fig. C.1. Mapping the unit circle by the transformation
$$u = b_{12} + b_{22} z_1.$$

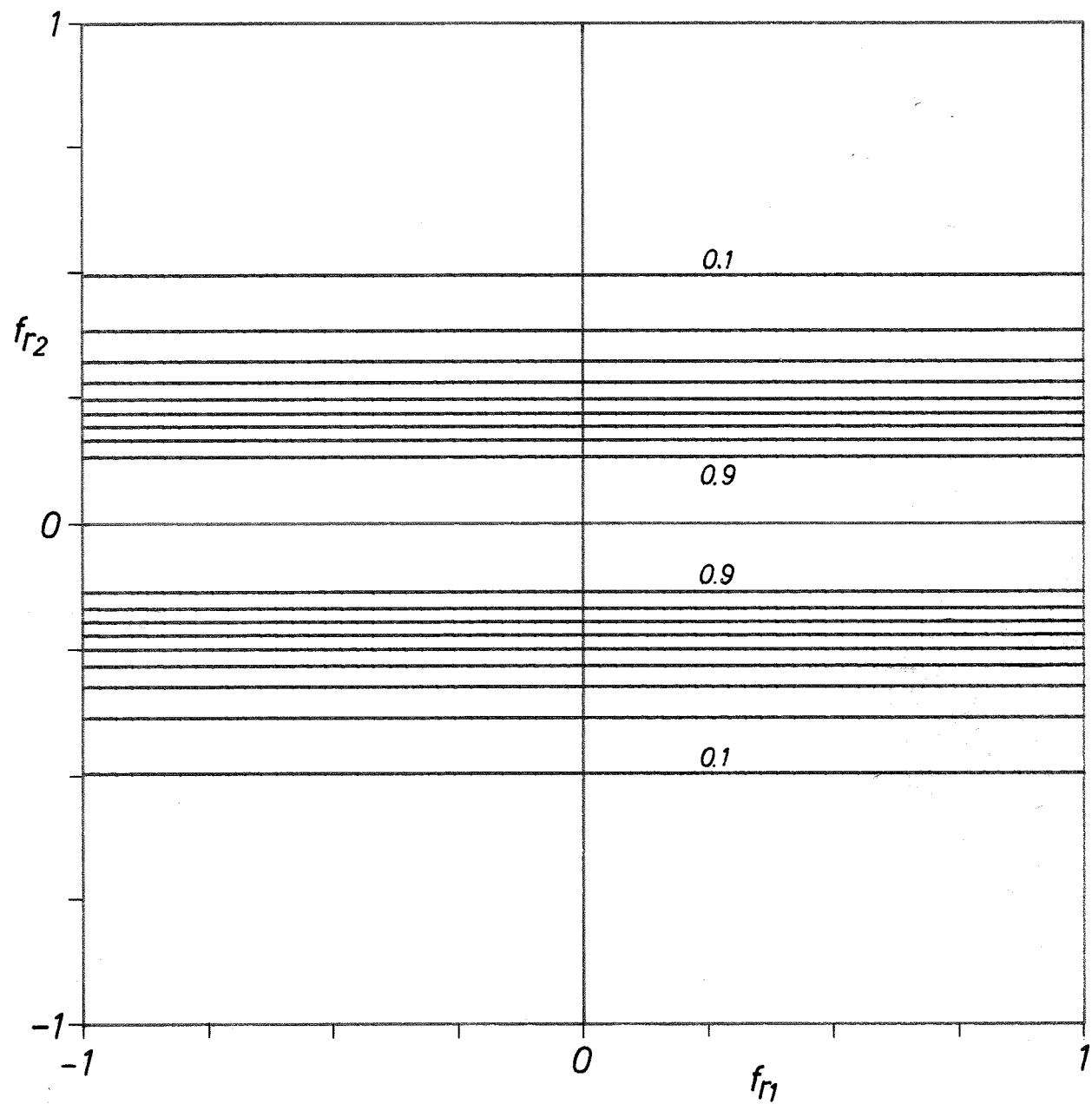


Fig. 1. Contour plot of the two-dimensional magnitude response
of a second-order Butterworth filter rotated 0° .

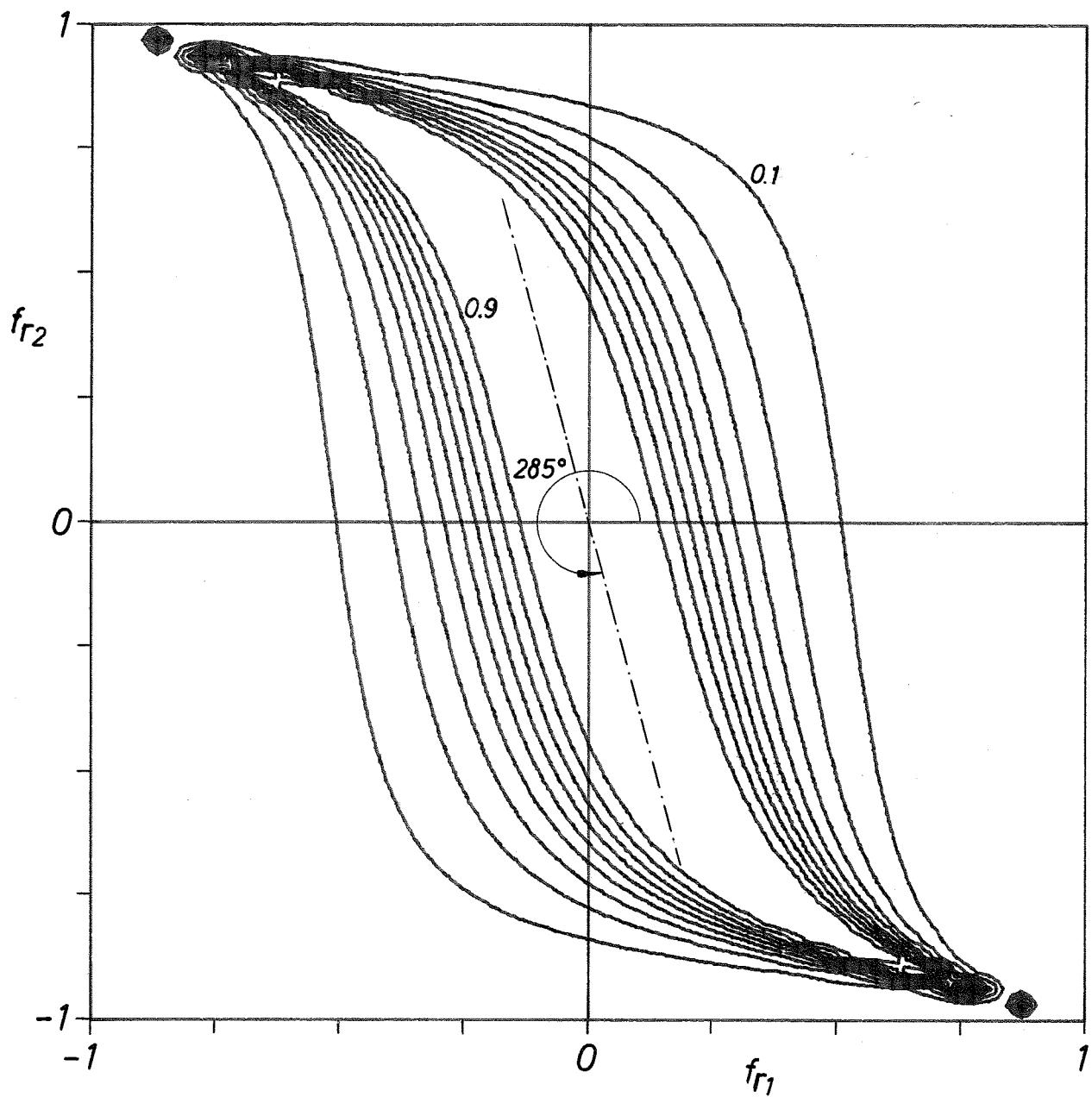


Fig. 2. Contour plot of the two-dimensional magnitude response
of a second-order Butterworth filter rotated 285° .

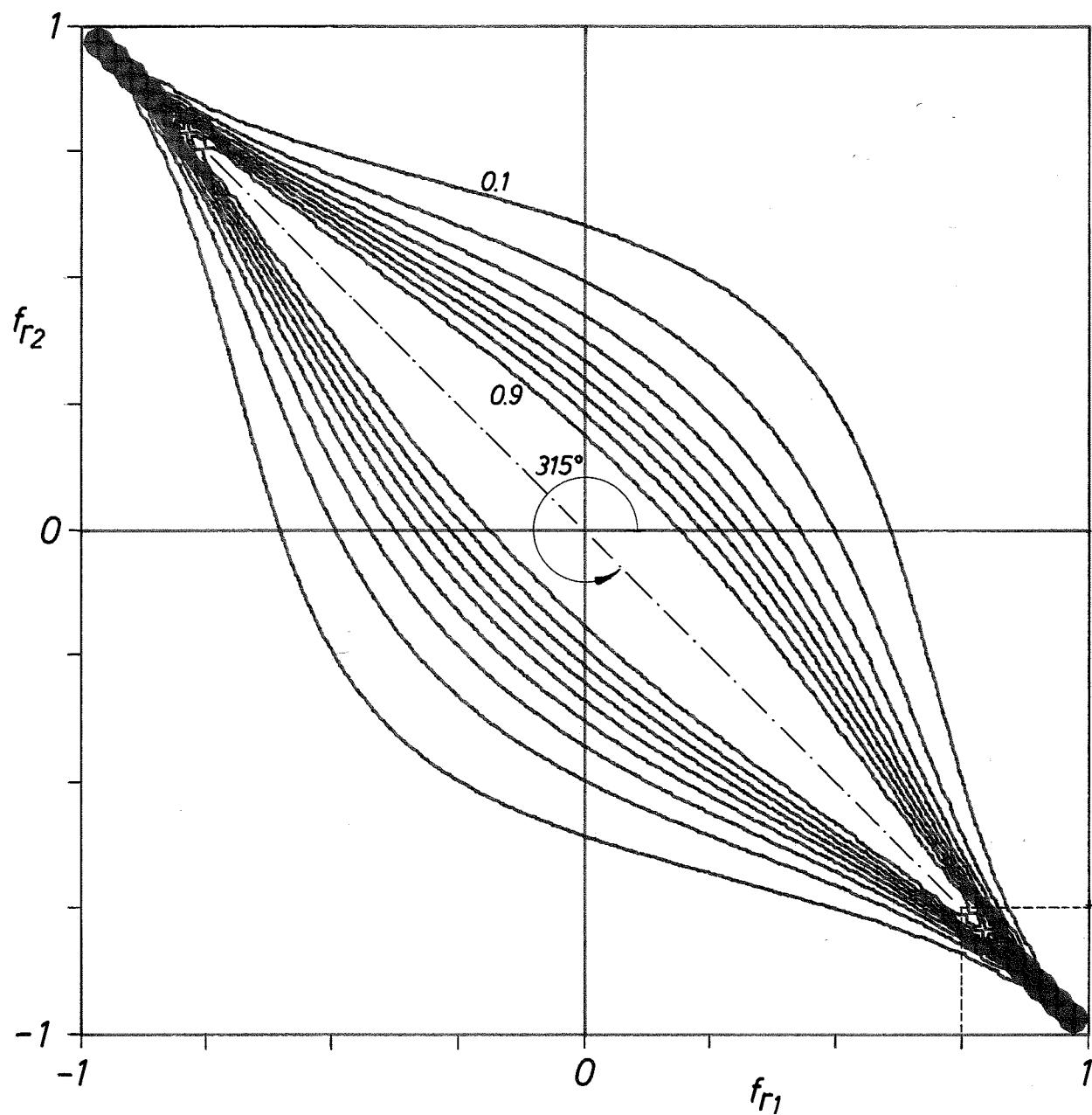


Fig. 3. Contour plot of the two-dimensional magnitude response
of a second-order Butterworth filter rotated 315° .

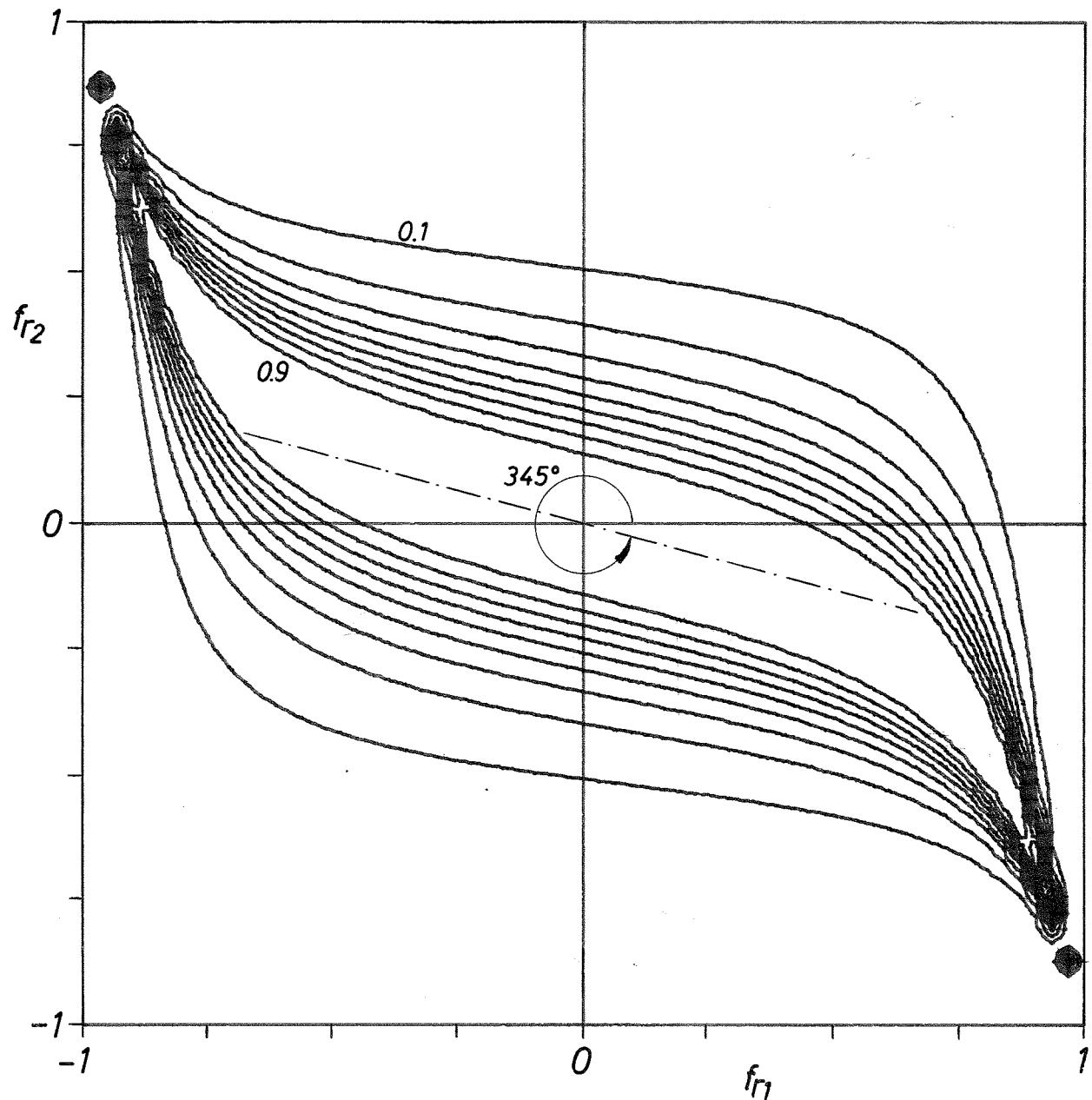
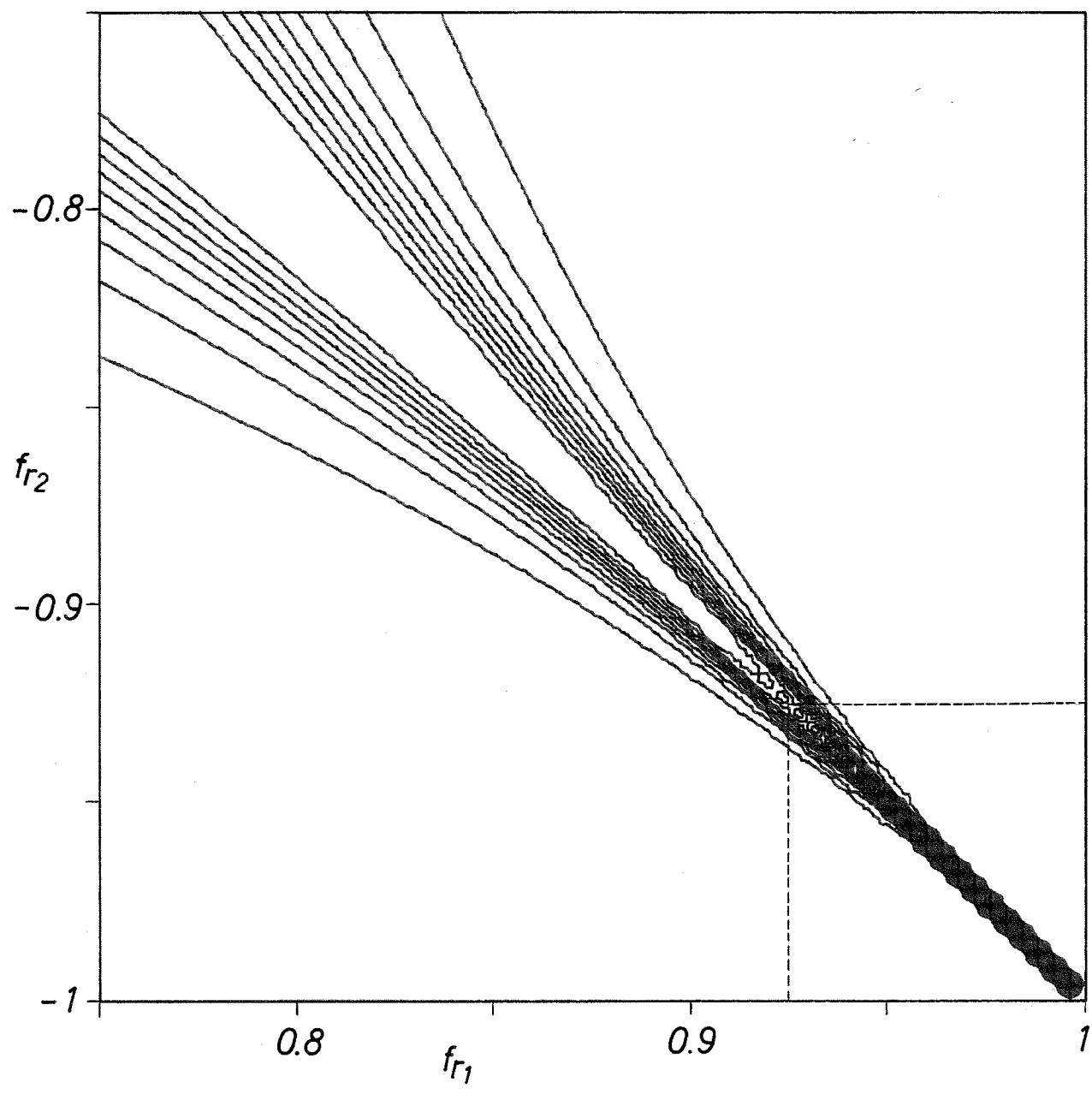
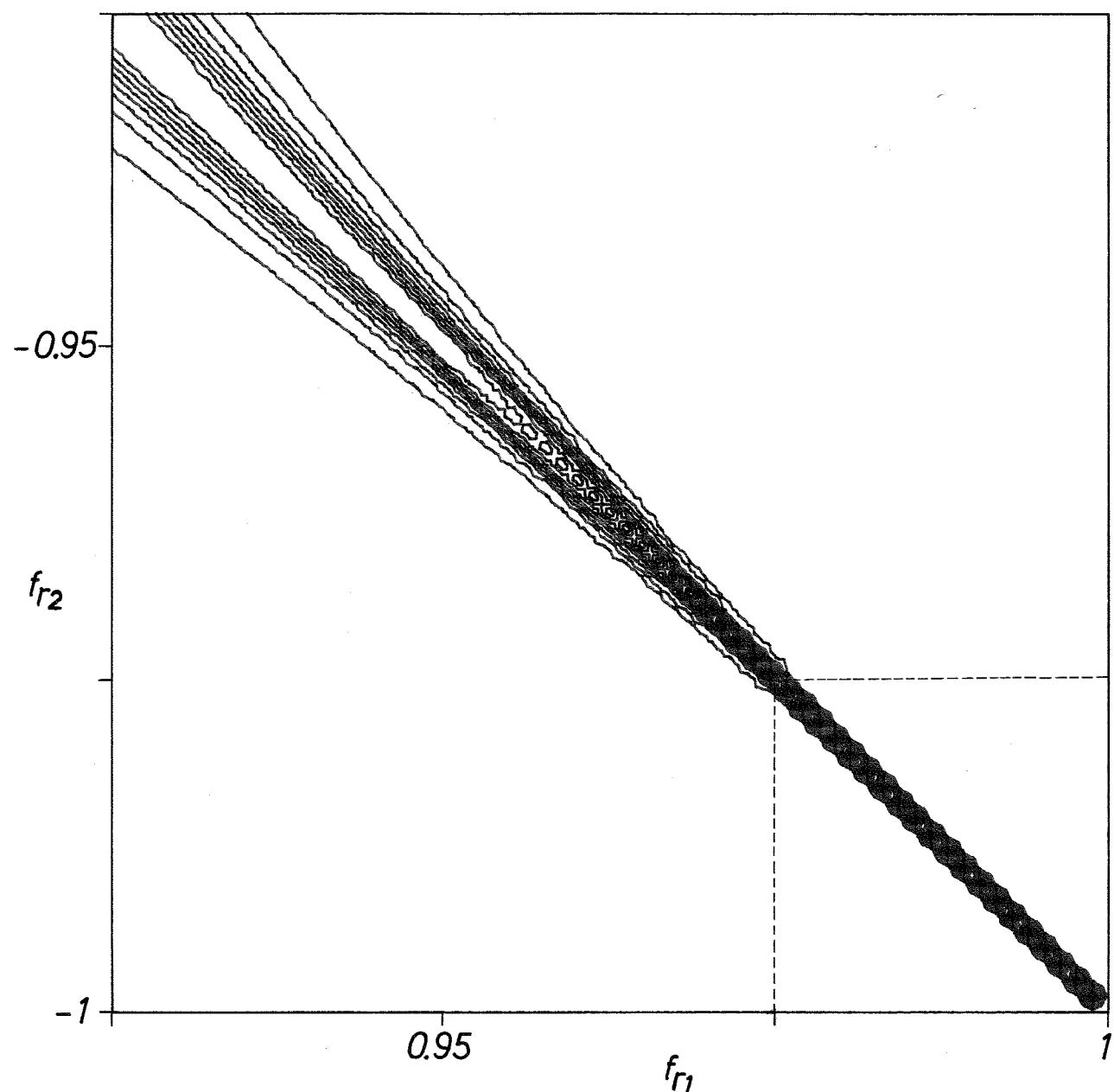


Fig. 4. Contour plot of the two-dimensional magnitude response
of a second-order Butterworth filter rotated 345° .



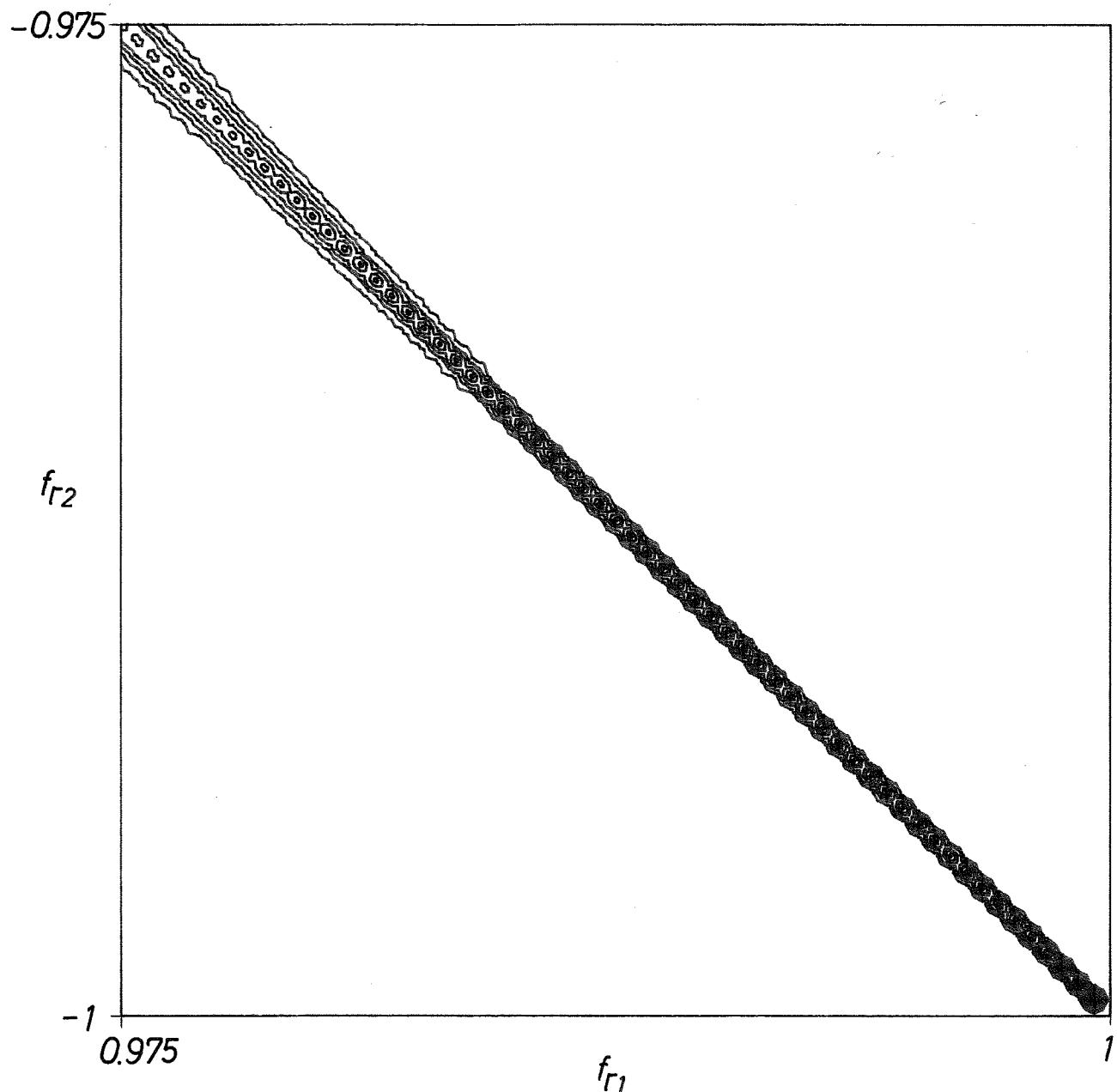
(a)

Fig. 5. Fine detail of the contour plot in Fig. 3 at high frequencies.



(b)

Fig. 5



(c)

Fig. 5

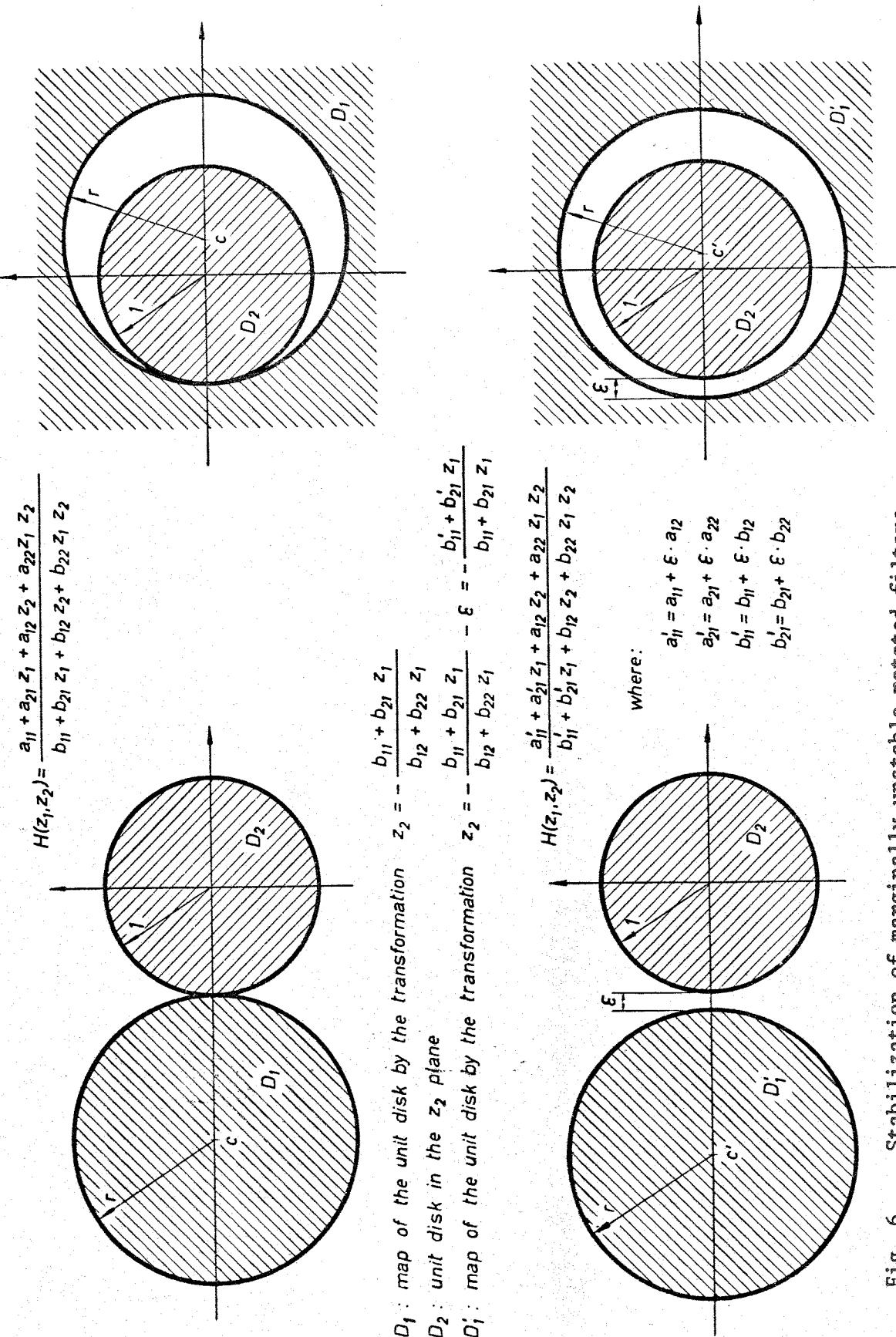
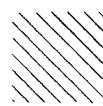
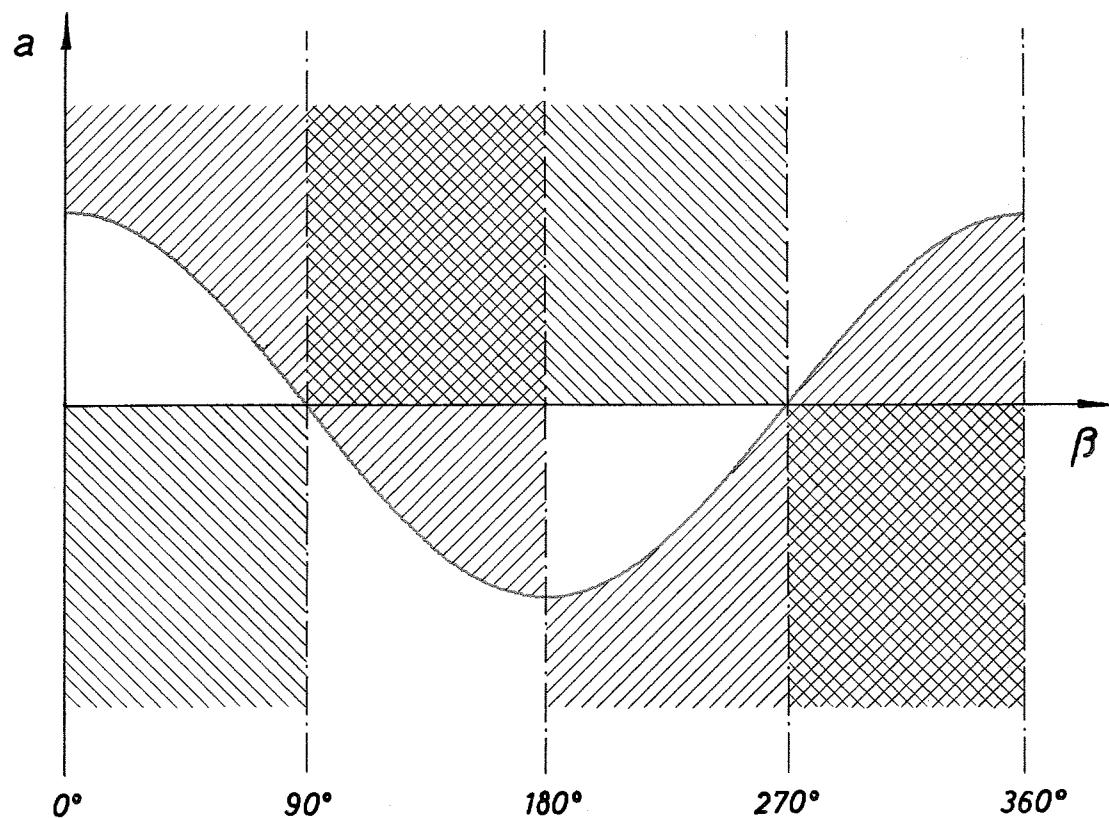
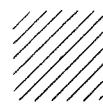


Fig. 6. Stabilization of marginally-unstable rotated filters.



Region of stability for the first condition.



Region of stability for the second condition.



Regions in which a rotated filter is stable.

Fig. 7. Regions of stability for rotated filters with transfer function $H(z_1, z_2)$.

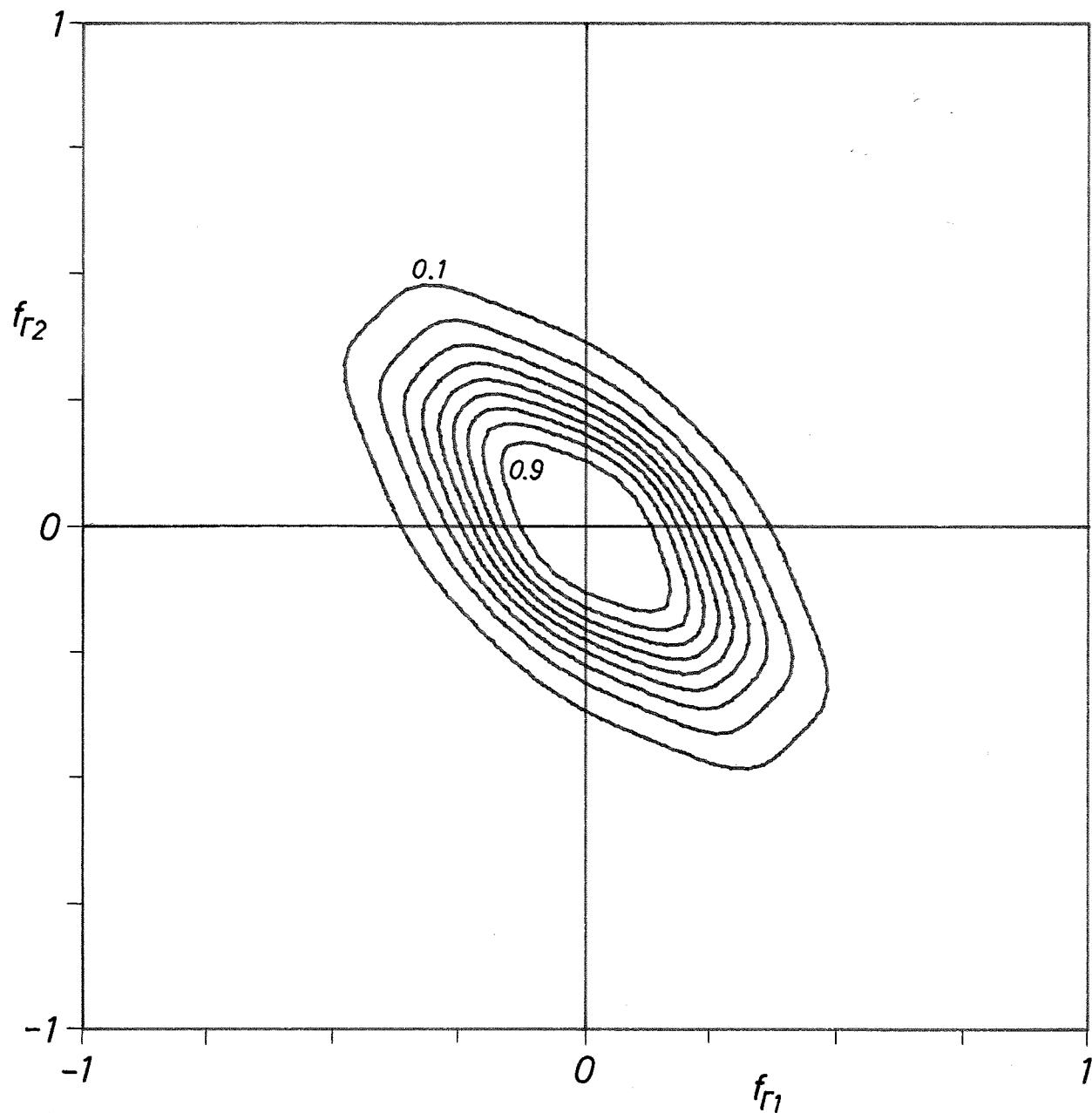


Fig. 8. Contour plot of the two-dimensional magnitude response of a cascade of three second-order Butterworth filters rotated 285° , 315° , and 345° .

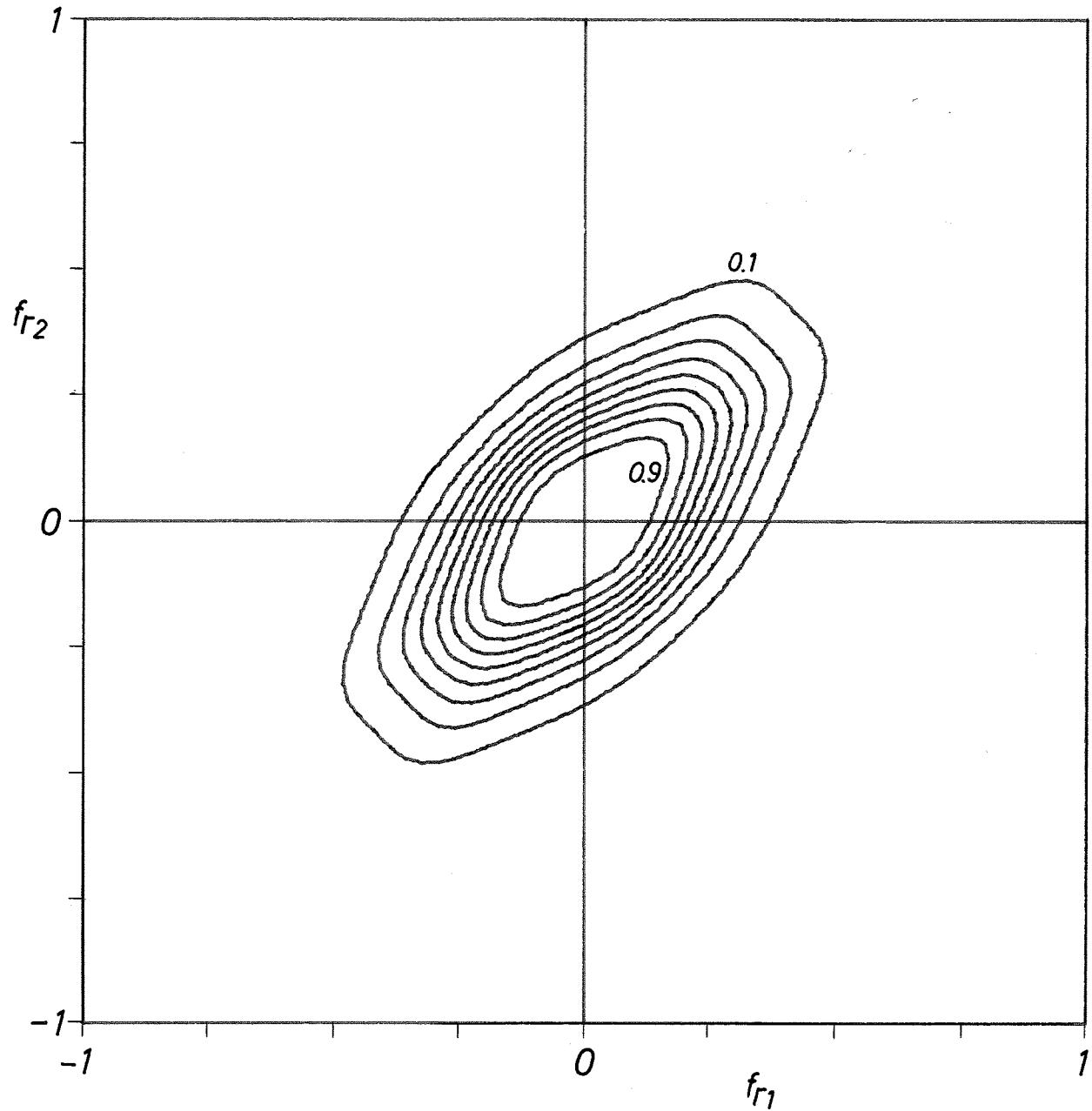


Fig. 9. Contour plot of the two-dimensional magnitude response of a cascade of three second-order Butterworth filters rotated 195° , 225° , and 255° .

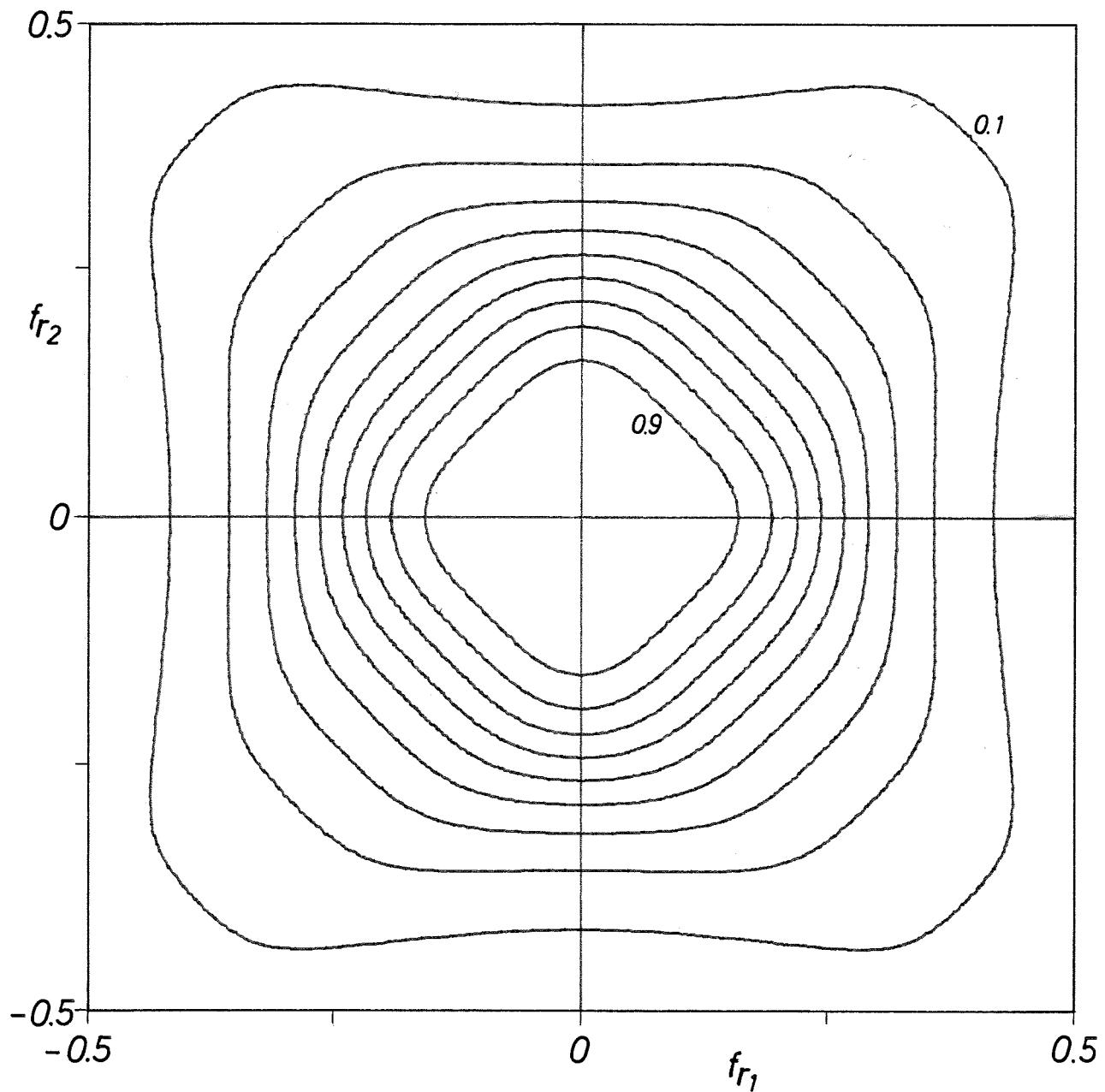


Fig. 10. Contour plot of the two-dimensional magnitude response of a cascade of two second-order Butterworth filters rotated 225° and 315° .

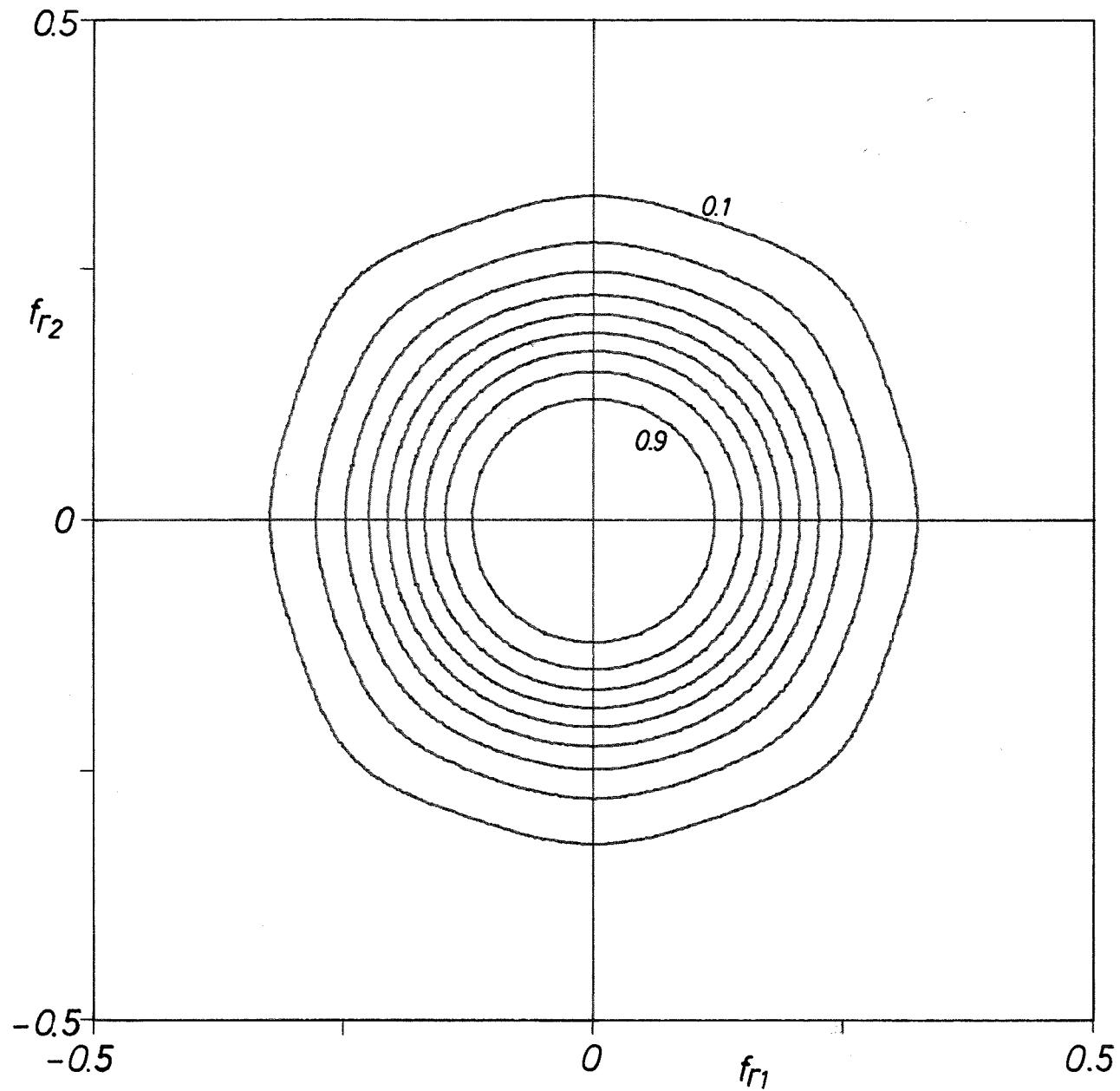


Fig. 11. Contour plot of the two-dimensional magnitude response
of a cascade of four second-order Butterworth filters
rotated by multiples of 45° .

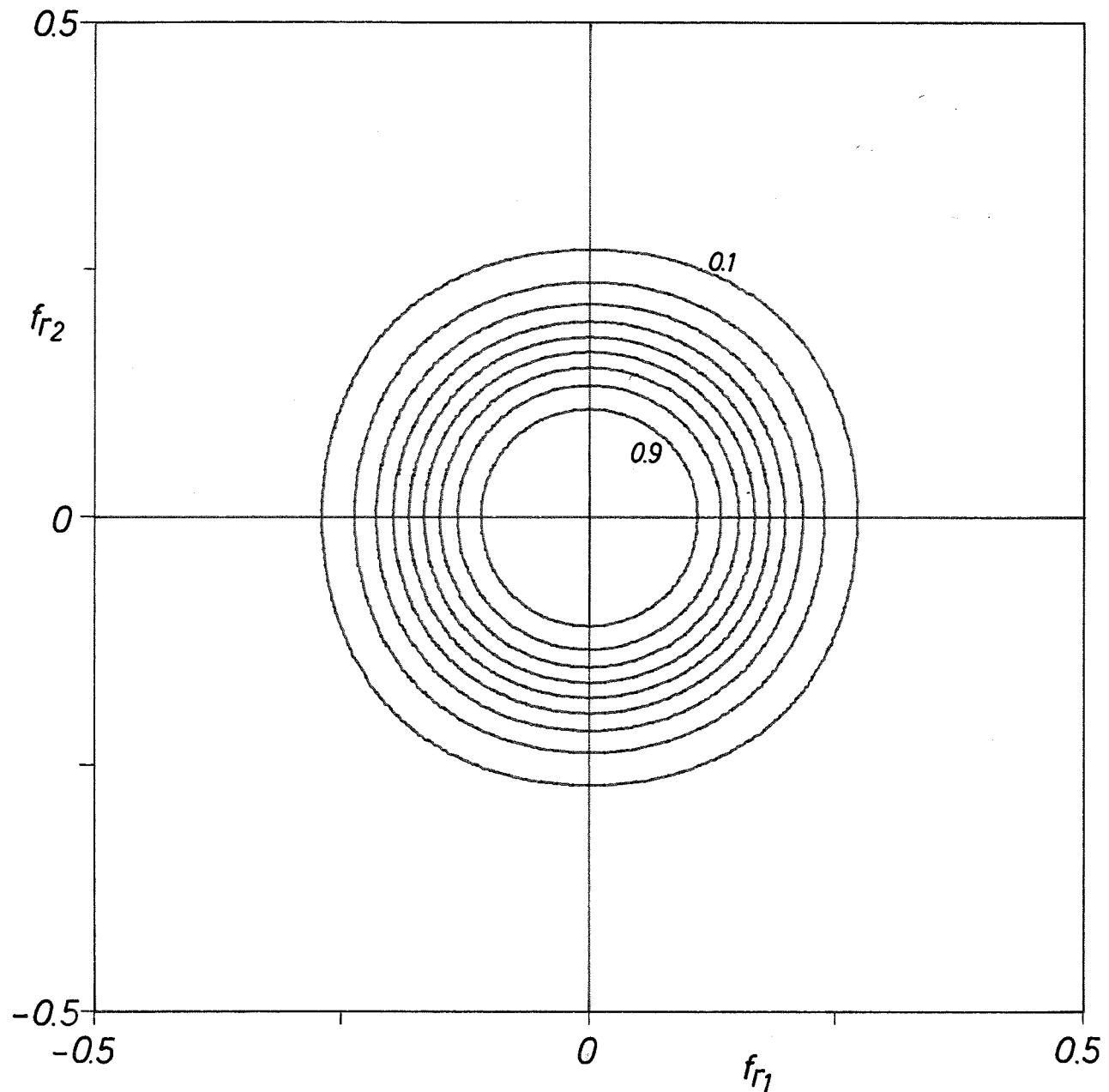


Fig. 12. Contour plot of the two-dimensional magnitude response
of a cascade of six second-order Butterworth filters
rotated by multiples of 30° .

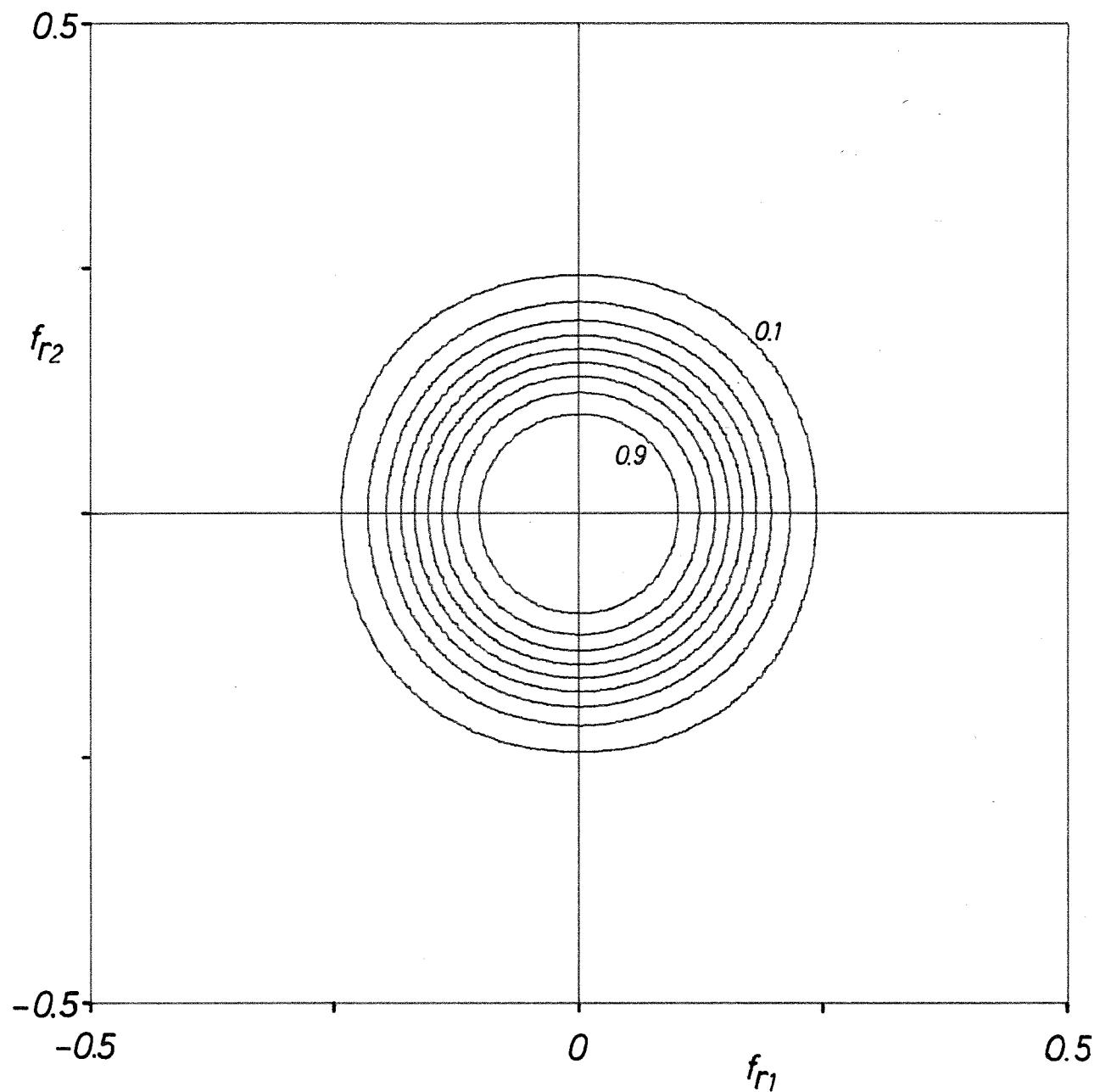


Fig. 13. Contour plot of the two-dimensional magnitude response of a cascade of eight second-order Butterworth filters rotated by multiples of 22.5° .

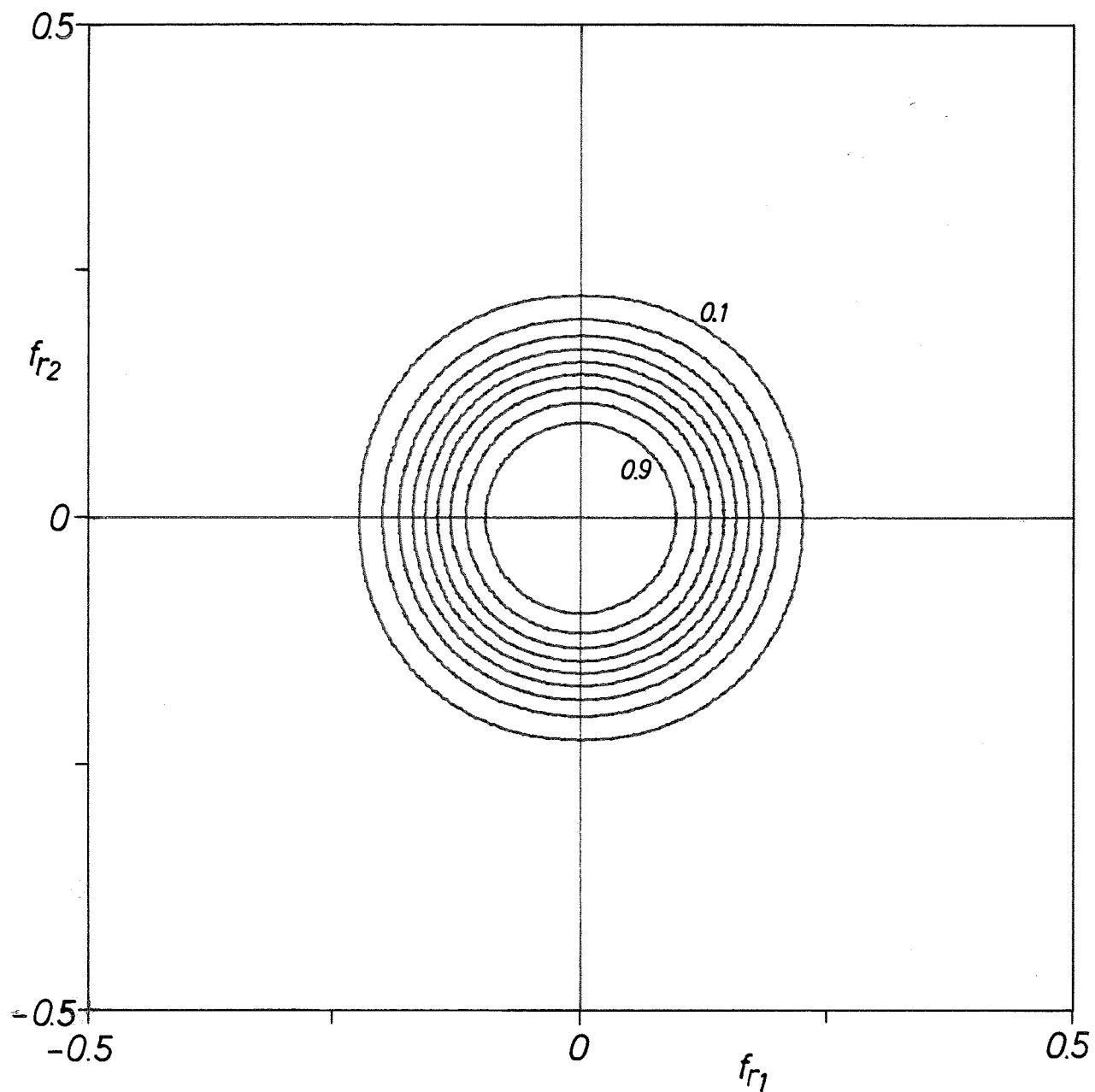


Fig. 14. Contour plot of the two-dimensional magnitude response of a cascade of ten second-order Butterworth filters rotated by multiples of 18° .

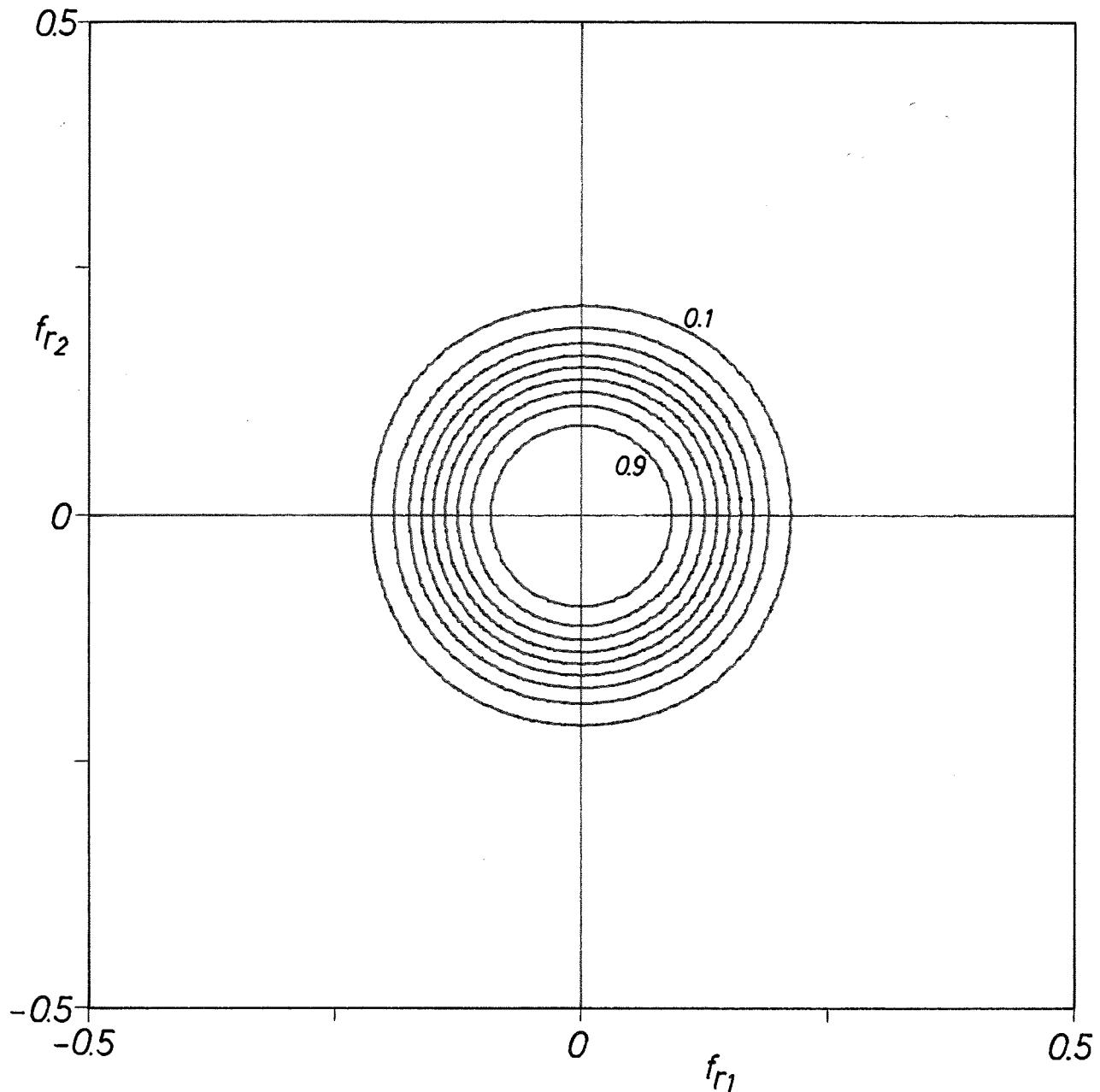


Fig. 15. Contour plot of the two-dimensional magnitude response of a cascade of twelve second-order Butterworth filters rotated by multiples of 15° .

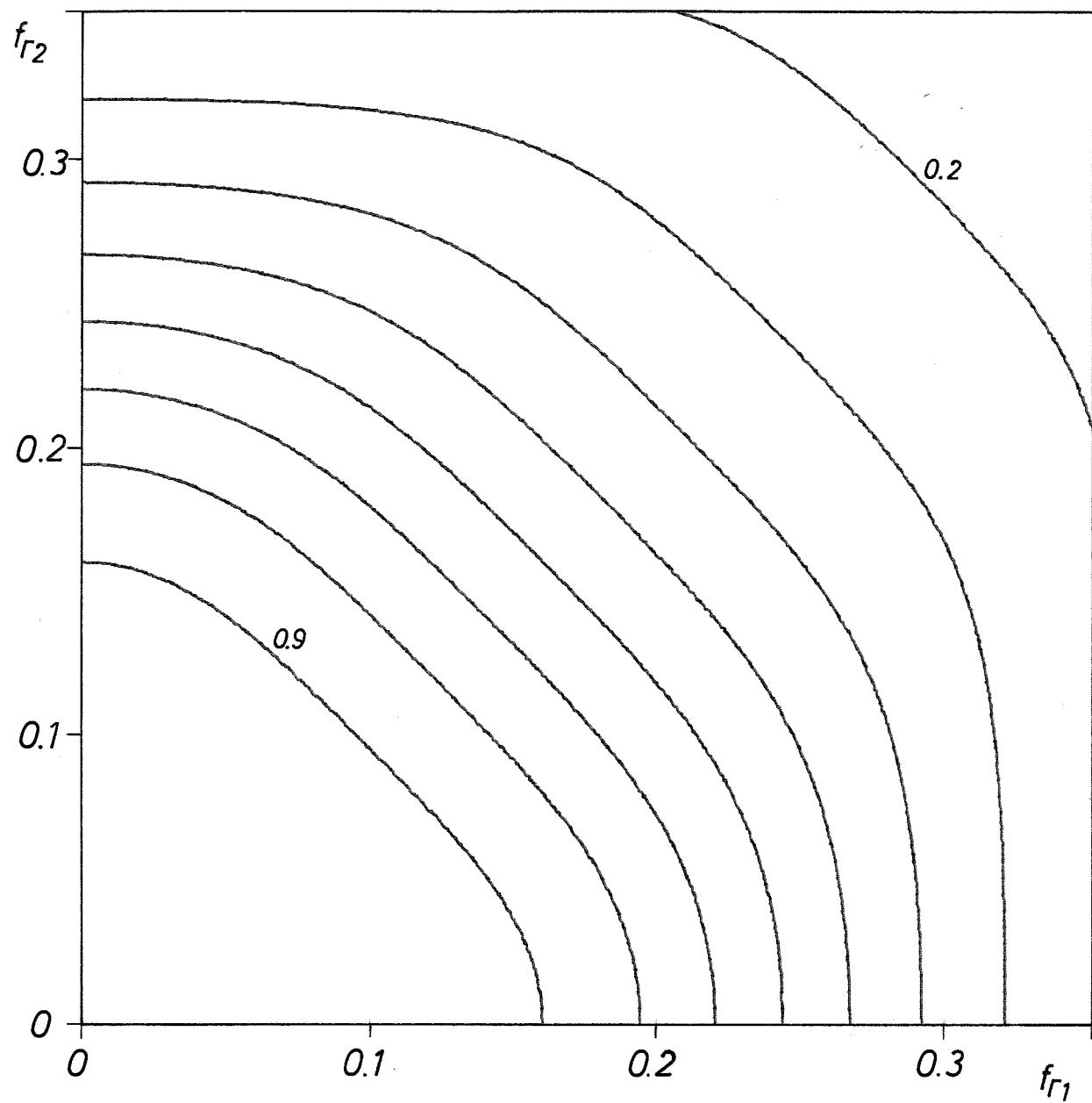


Fig. 16. Fine detail of one quadrant of the contour plot in Fig. 10.

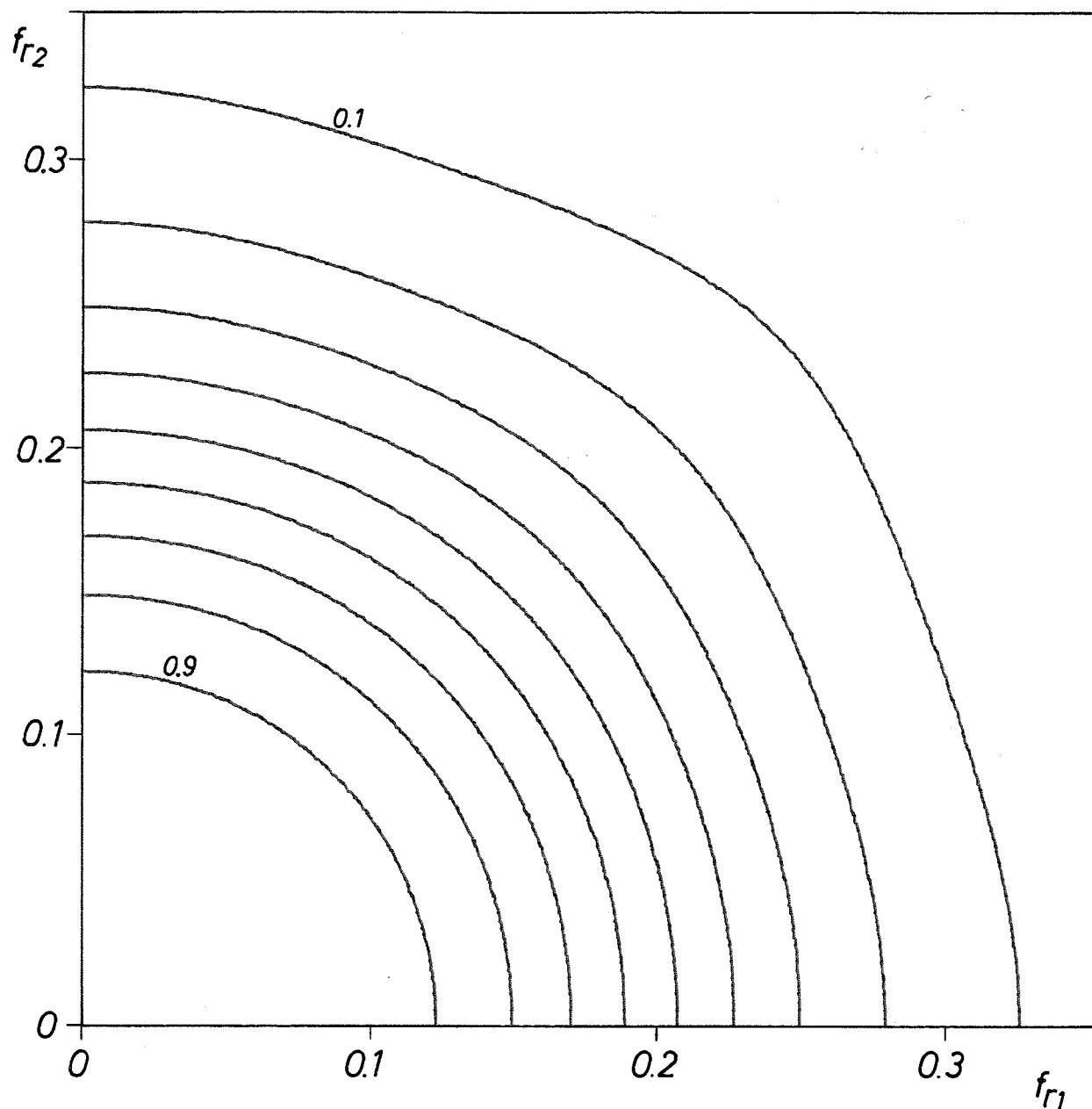


Fig. 17. Fine detail of one quadrant of the contour plot in Fig. 11.

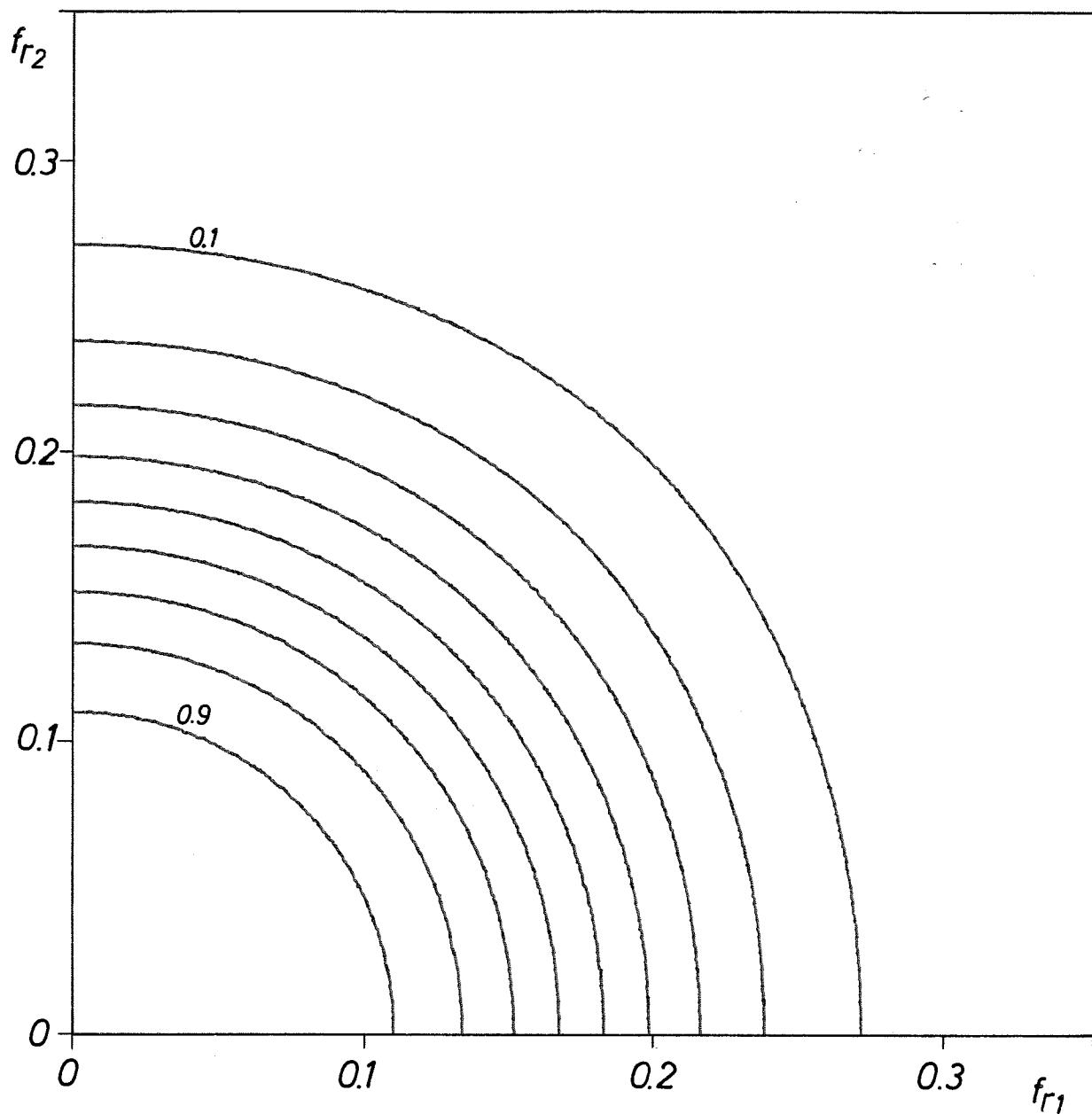


Fig. 18. Fine detail of one quadrant of the contour plot in Fig. 12.

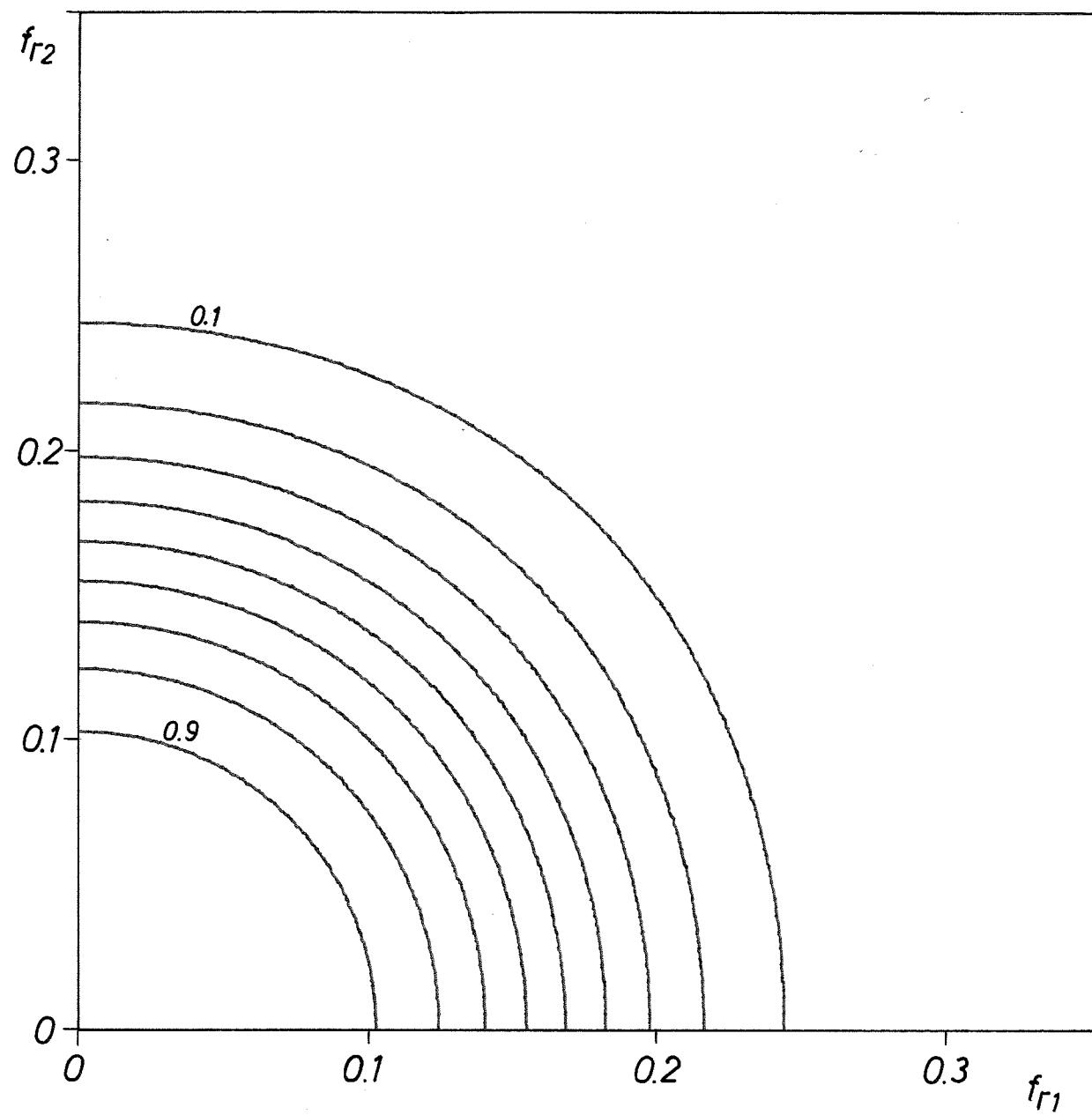


Fig. 19. Fine detail of one quadrant of the contour plot in Fig. 13.

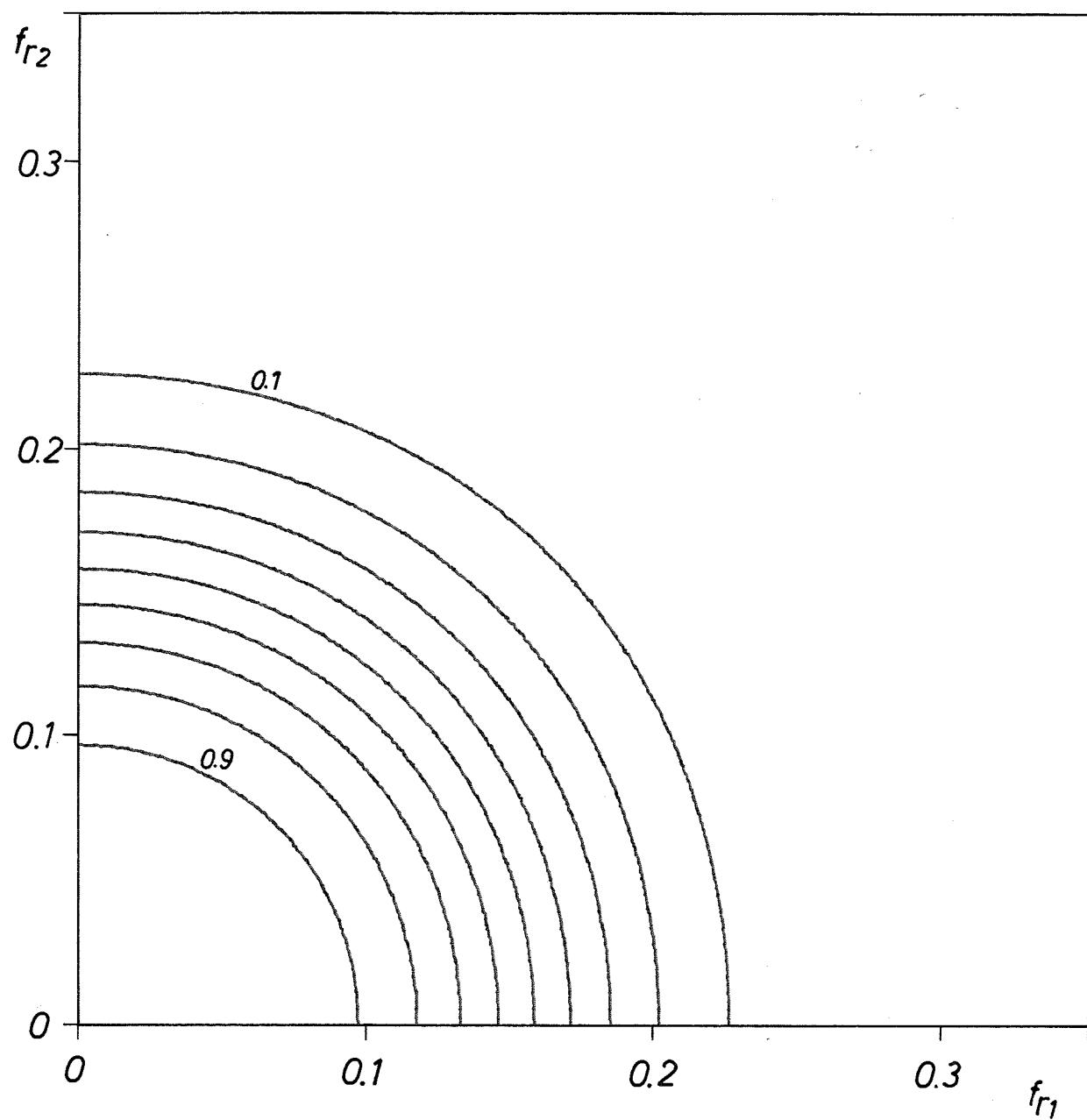


Fig. 20. Fine detail of one quadrant of the contour plot in Fig. 14.

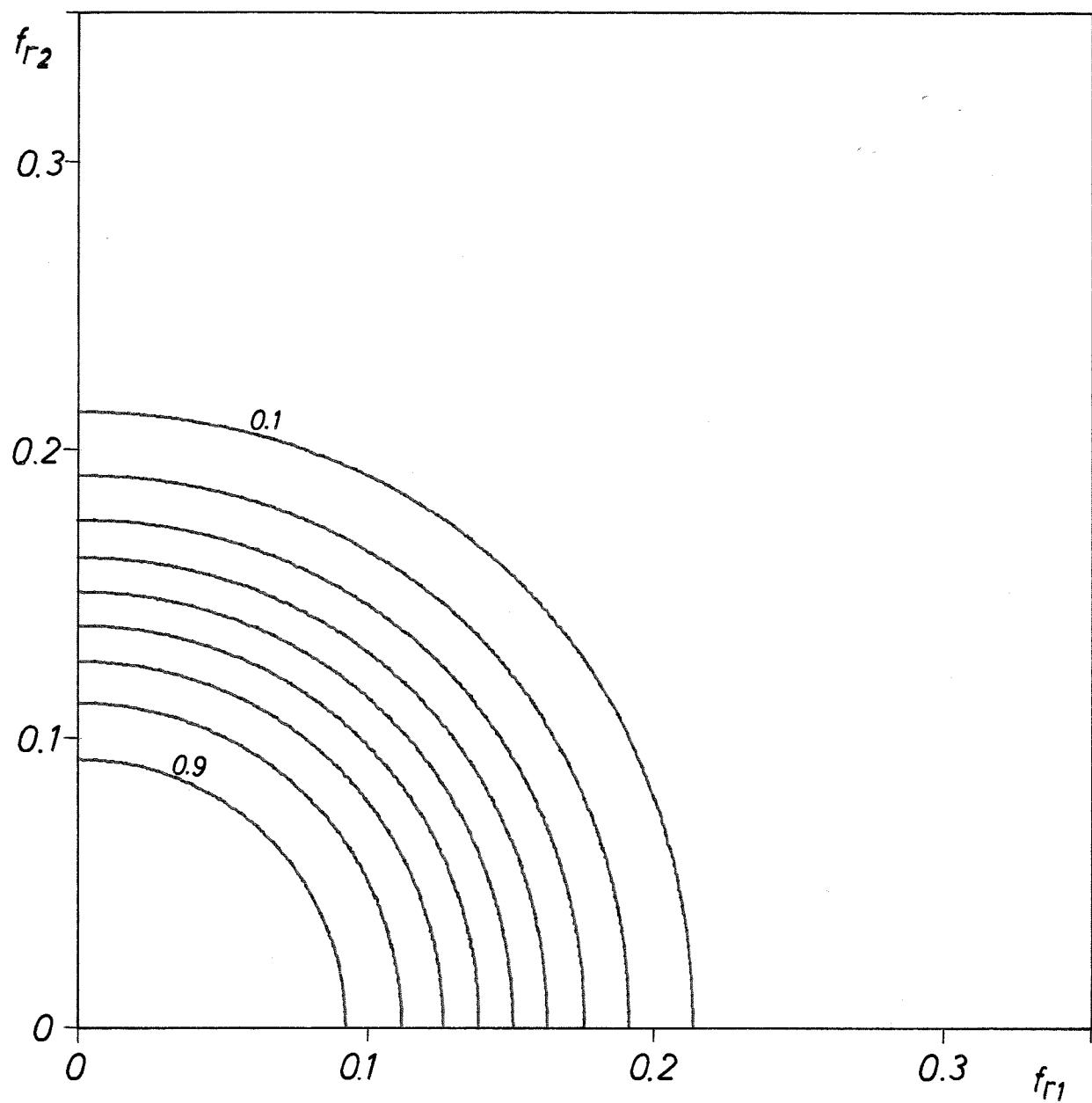


Fig. 21. Fine detail of one quadrant of the contour plot in Fig. 15.

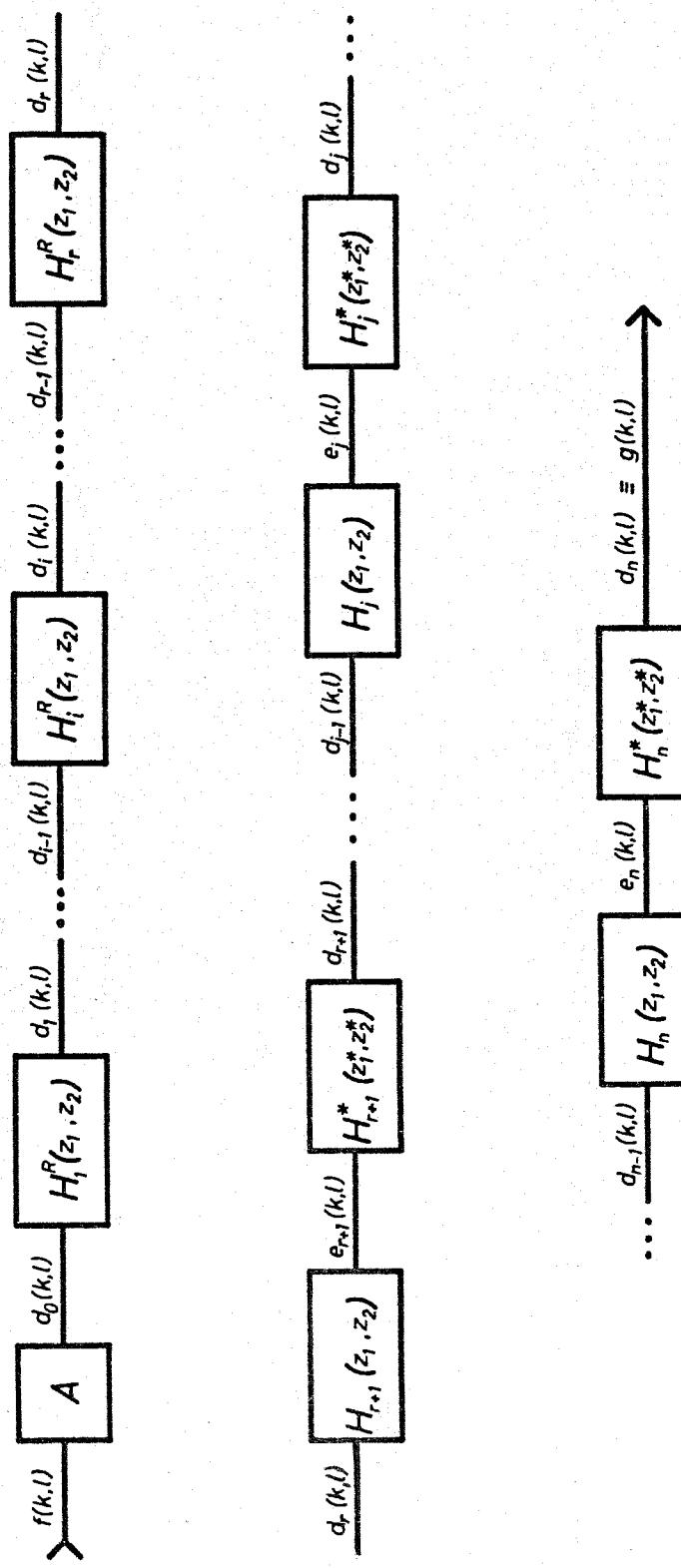
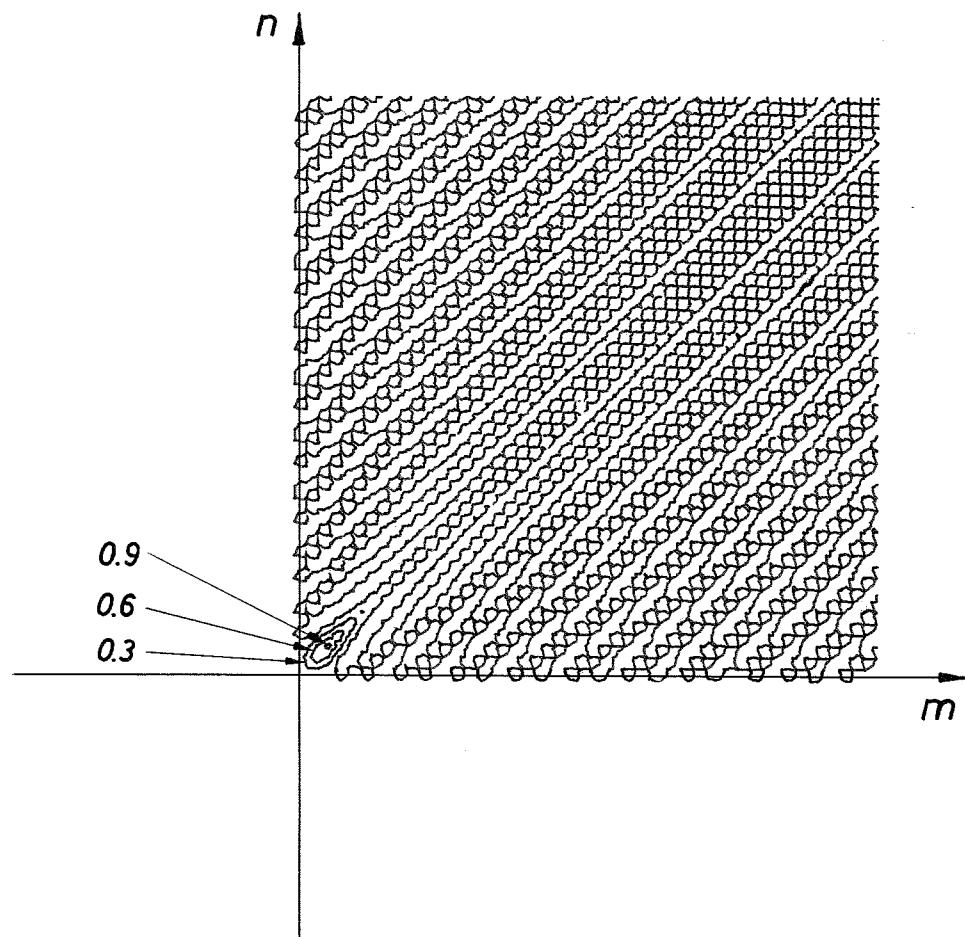


Fig. 22. Block diagram for two-dimensional complex cascade programming.



The unlabelled contours are at level 0.0

Fig. 23. Contour map of the impulse response of a second-order Butterworth filter rotated 315° .

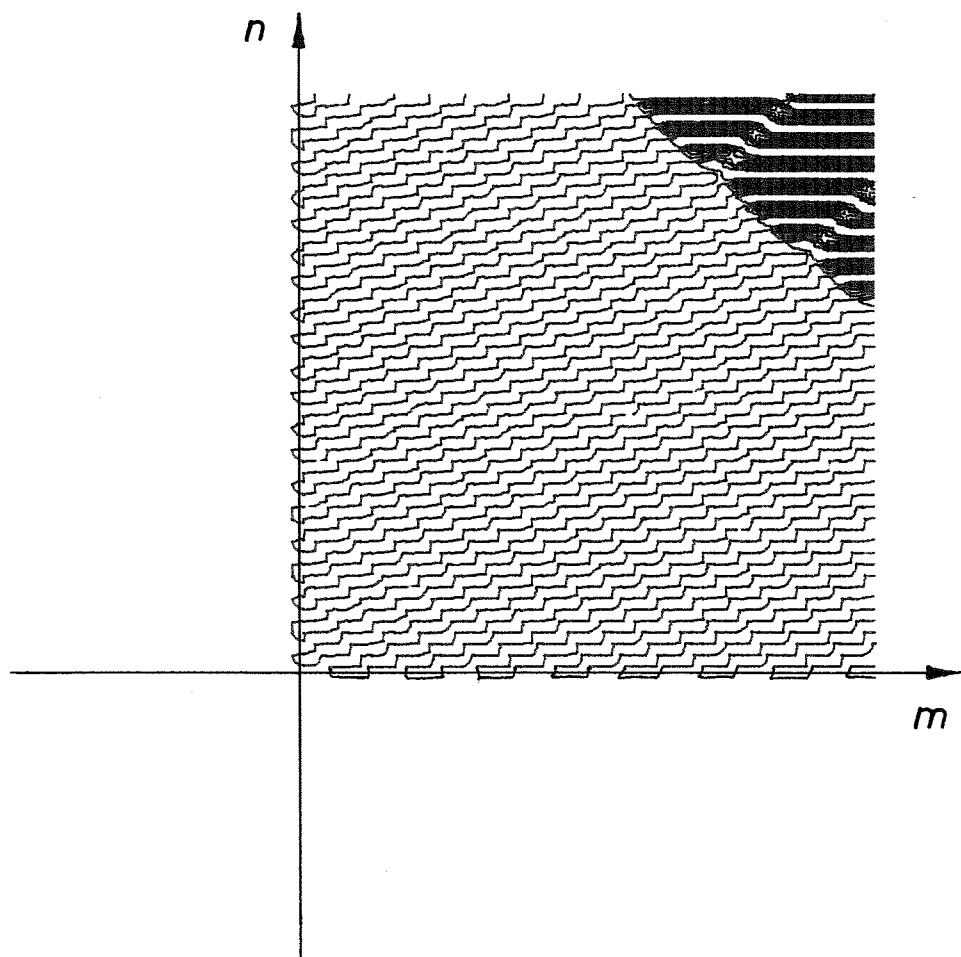
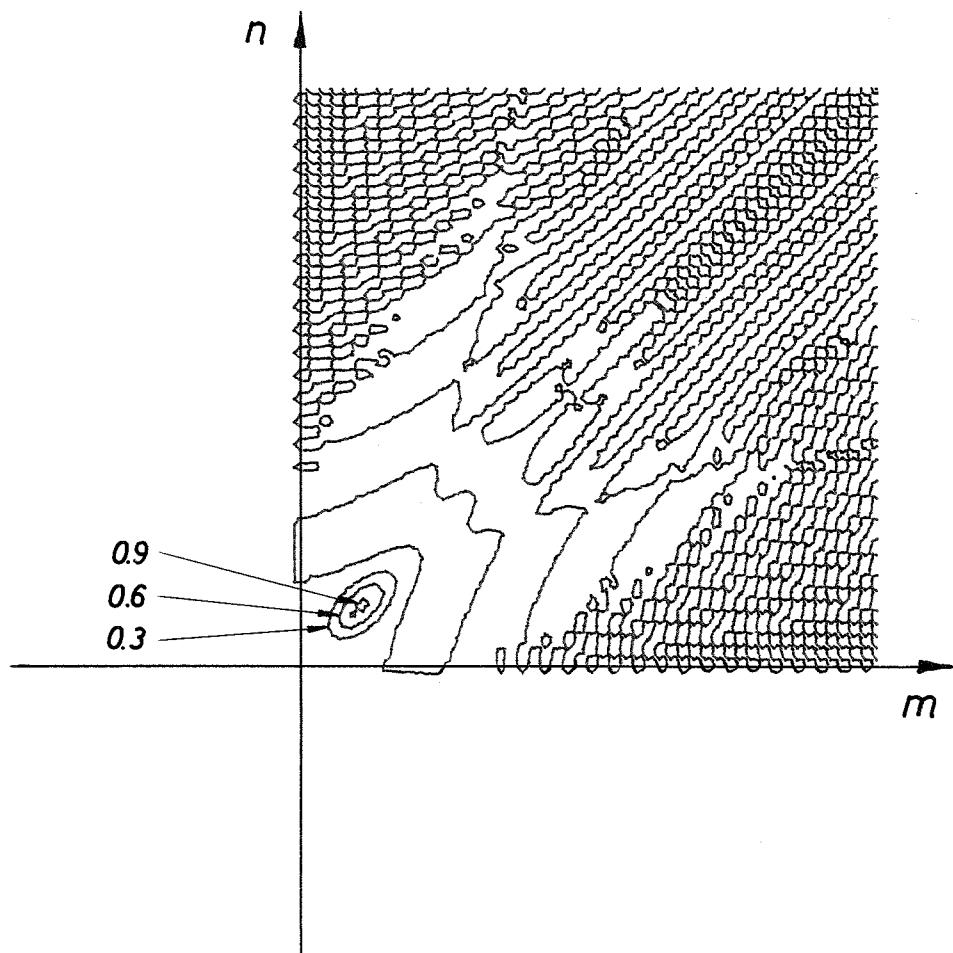
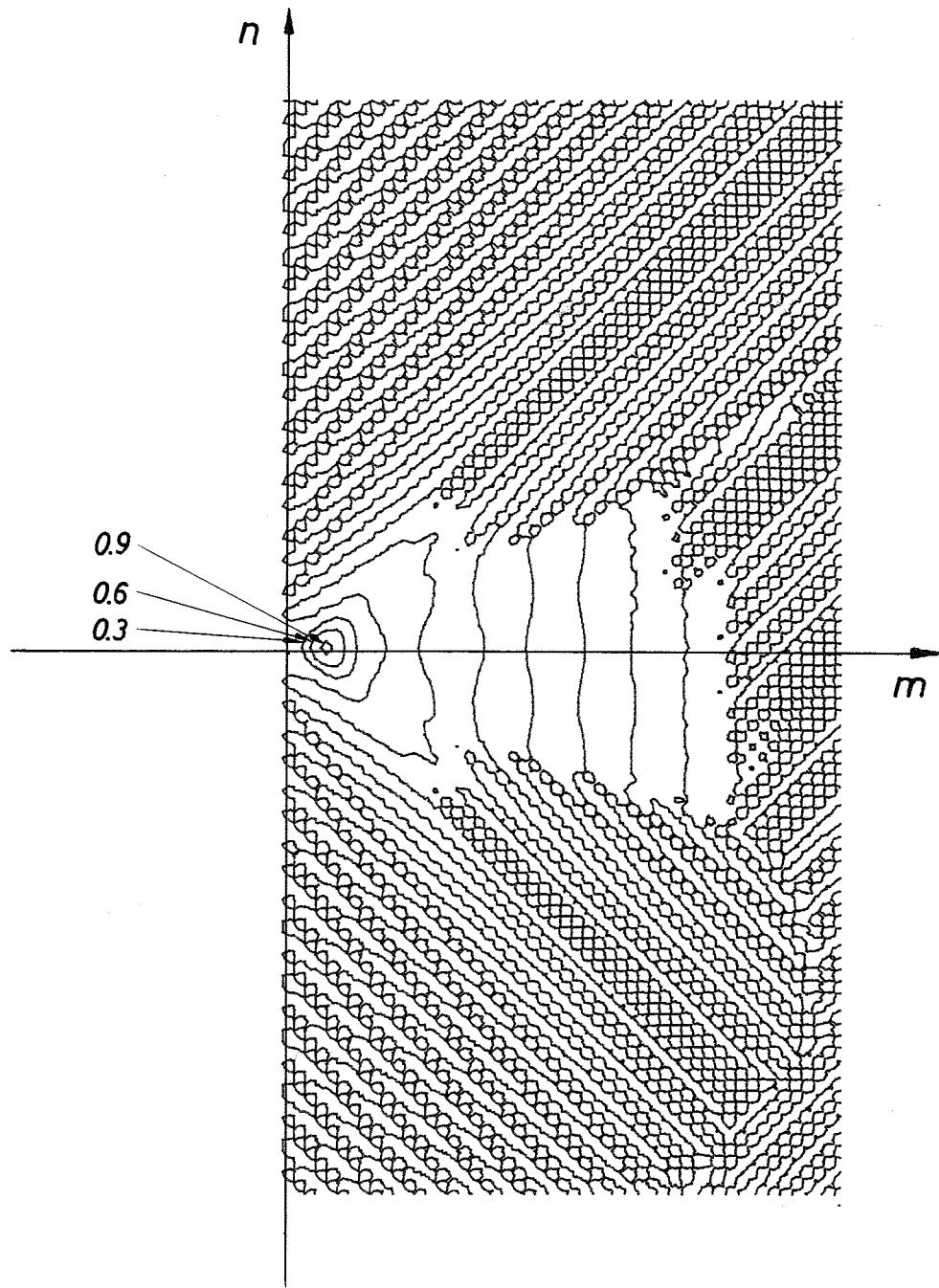


Fig. 24. Contour map of the impulse response of a second-order Butterworth filter rotated 225° .



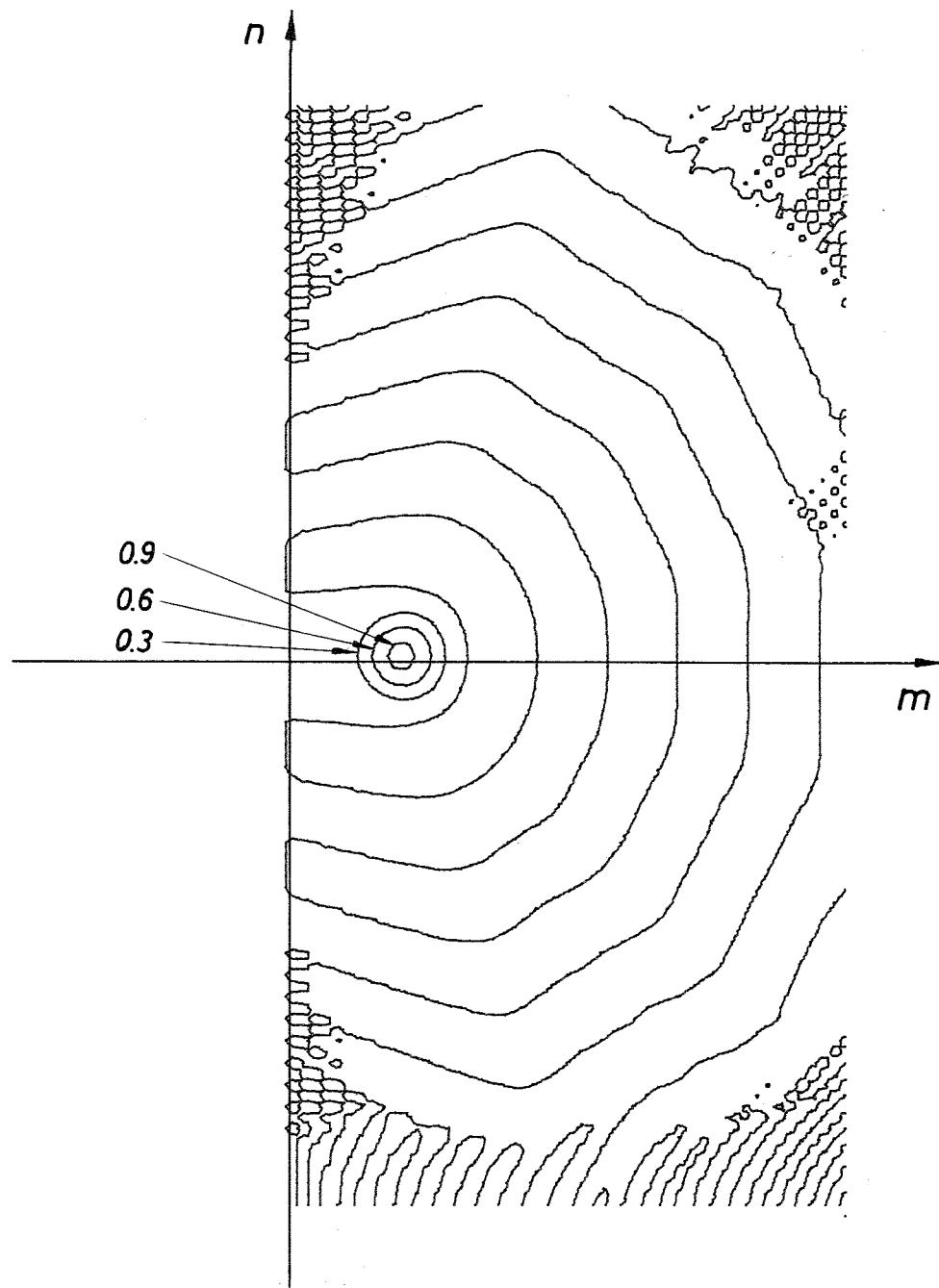
The unlabelled contours are at level 0.0

Fig. 25. Contour map of the impulse response of the filter shown in Fig. 8.



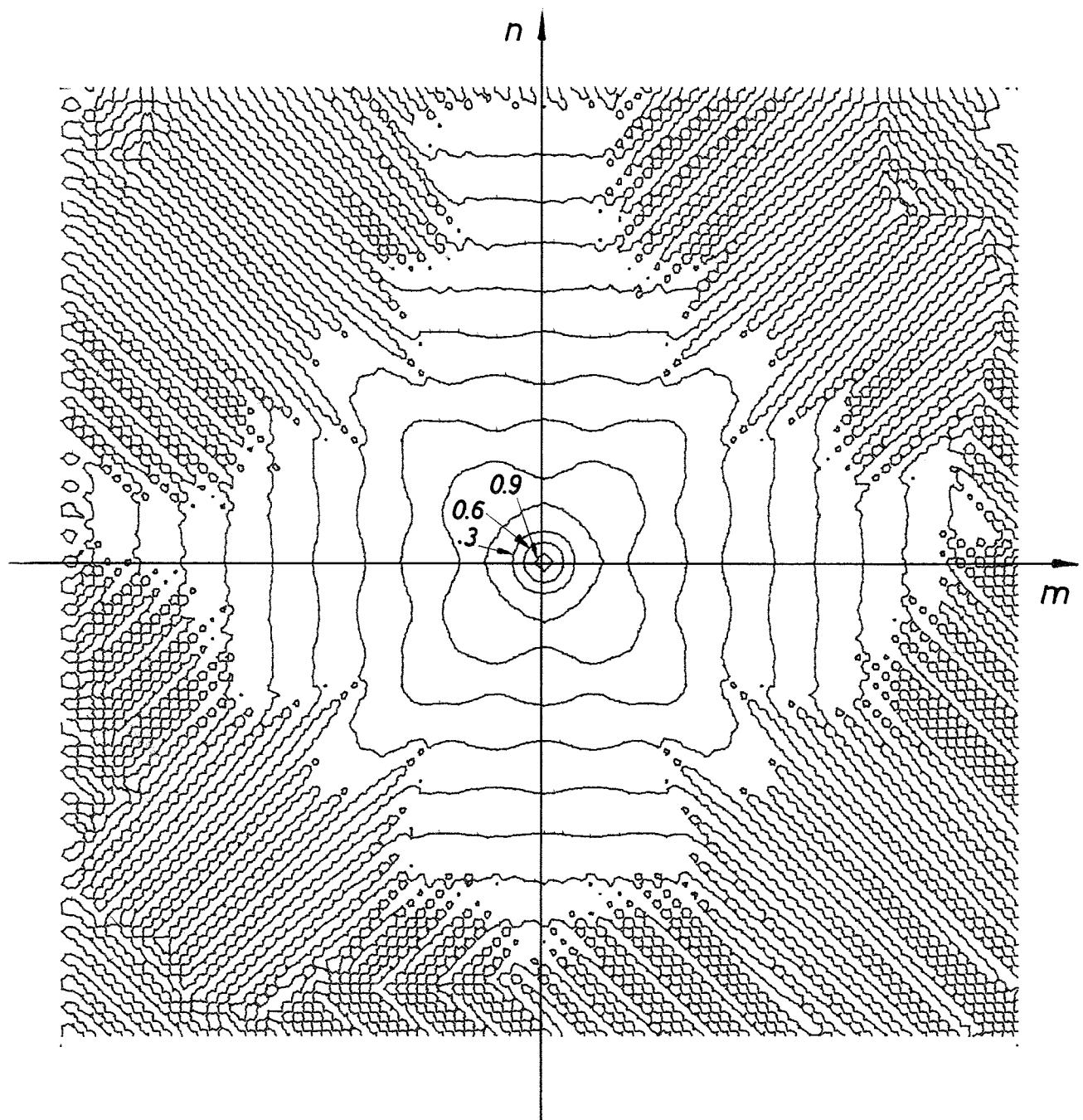
The unlabelled contours are at level 0.0

Fig. 26. Contour map of the impulse response of the filter shown in Fig. 10.



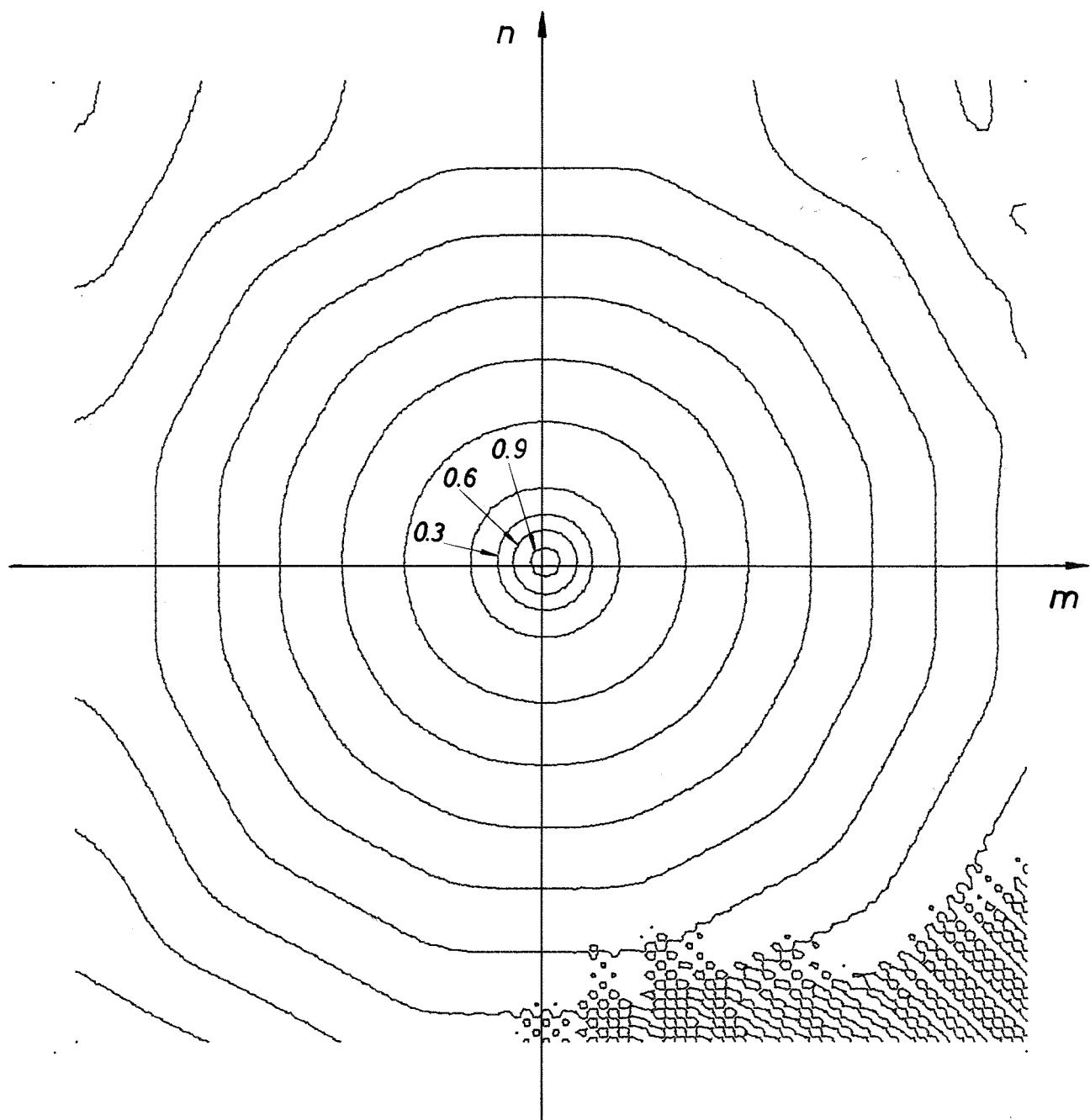
The unlabelled contours are at level 0.0

Fig. 27. Contour map of the impulse response of the filter shown in Fig. 12.



The unlabelled contours are at level 0.0

Fig. 28. Contour map of the impulse response of a filter with zero-phase response and magnitude response the square of the magnitude response shown in Fig. 10.



The unlabelled contours are at level 0.0

Fig. 29. Contour map of the impulse response of a filter with zero-phase response and magnitude response the square of the magnitude response shown in Fig. 12.

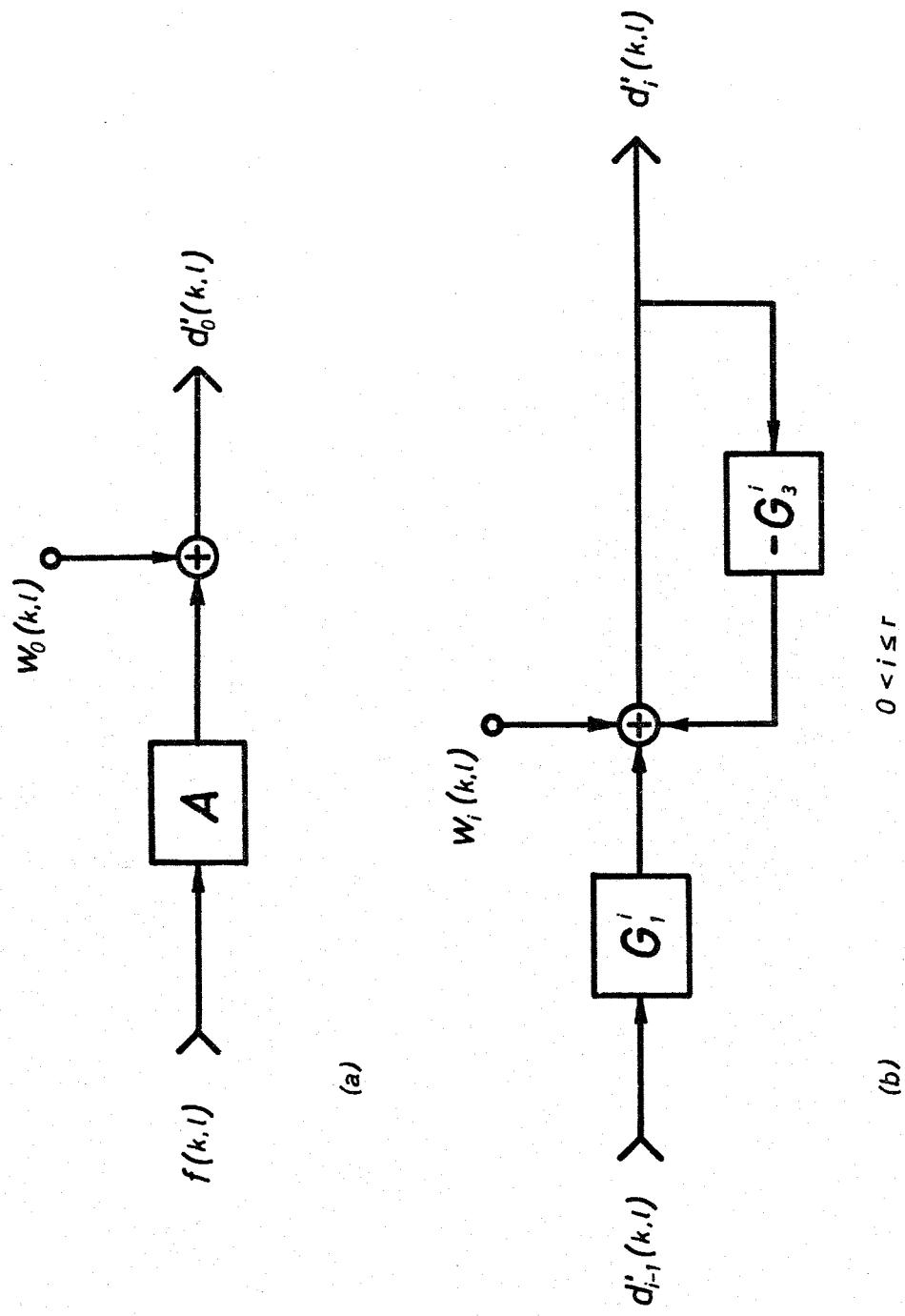


Fig. 30. Block diagrams for the elements in Fig. 22 including the computational errors.

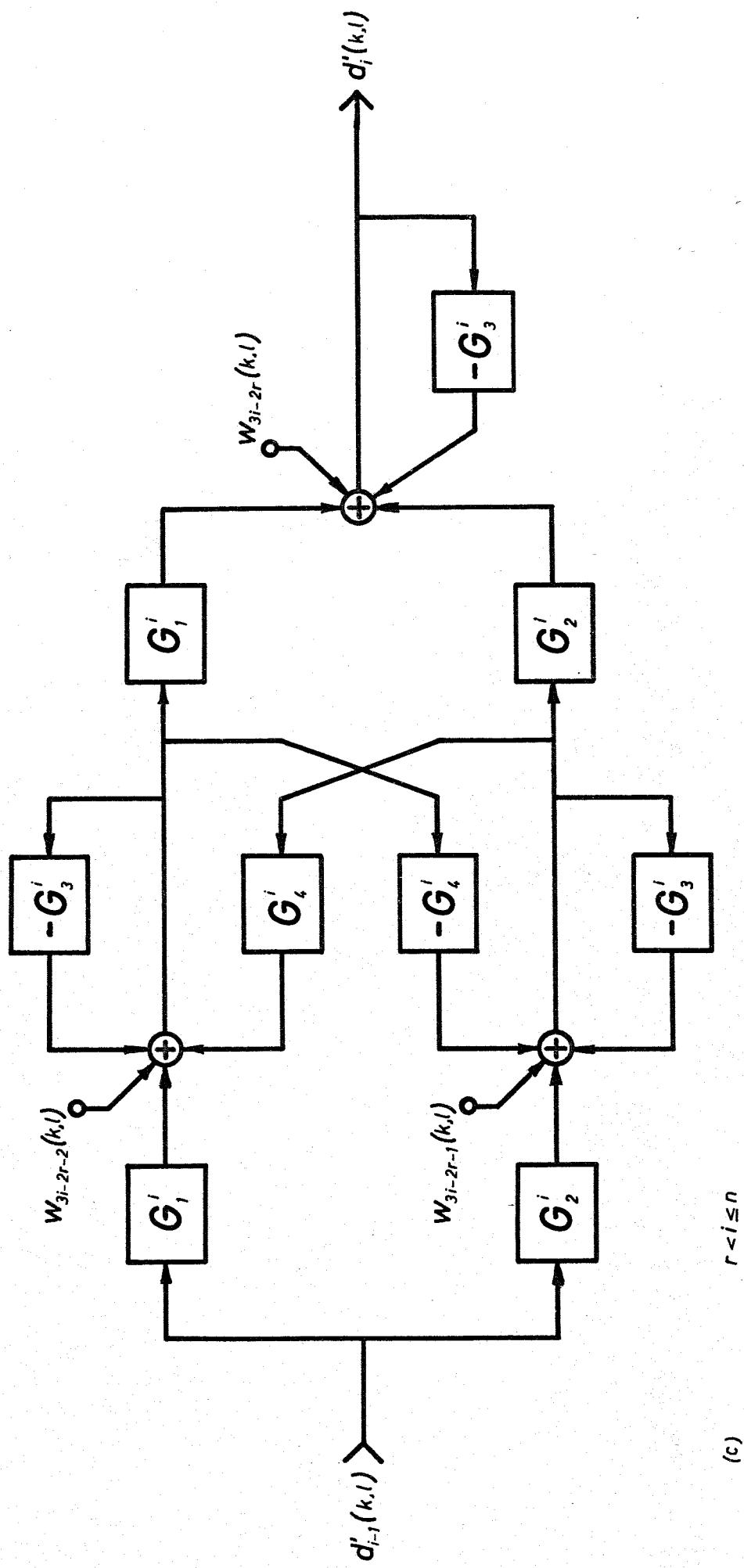


Fig. 30.

(c)

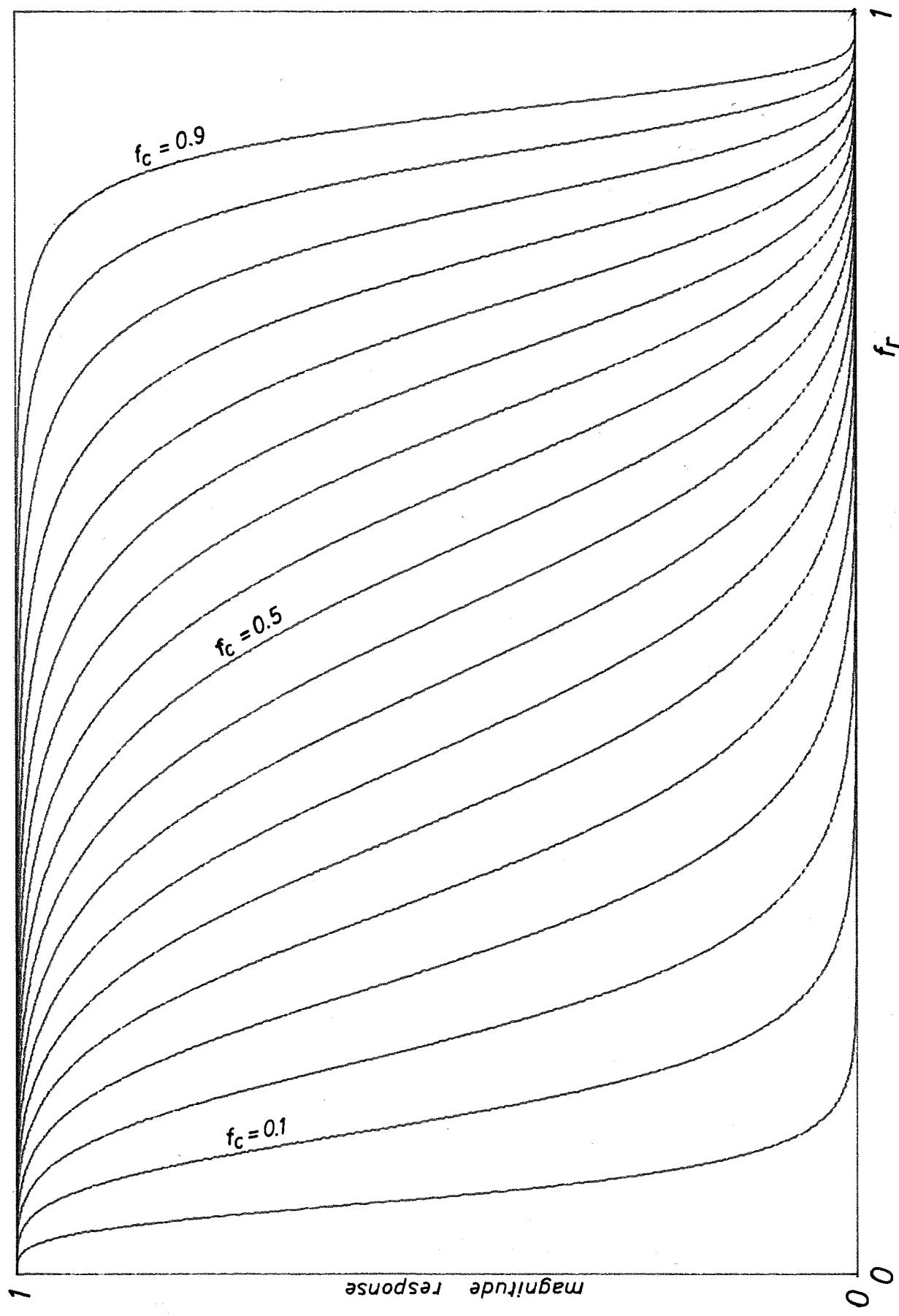


Fig. 32.

Magnitude responses, in the direction 0° , of 18 two-dimensional recursive filters with cutoff frequencies from 0.05 to 0.9 at intervals of 0.05. Each filter consists of a cascade of four second-order Butterworth filters rotated by 202.5° , 247.5° , 292.5° , and 337.5° .

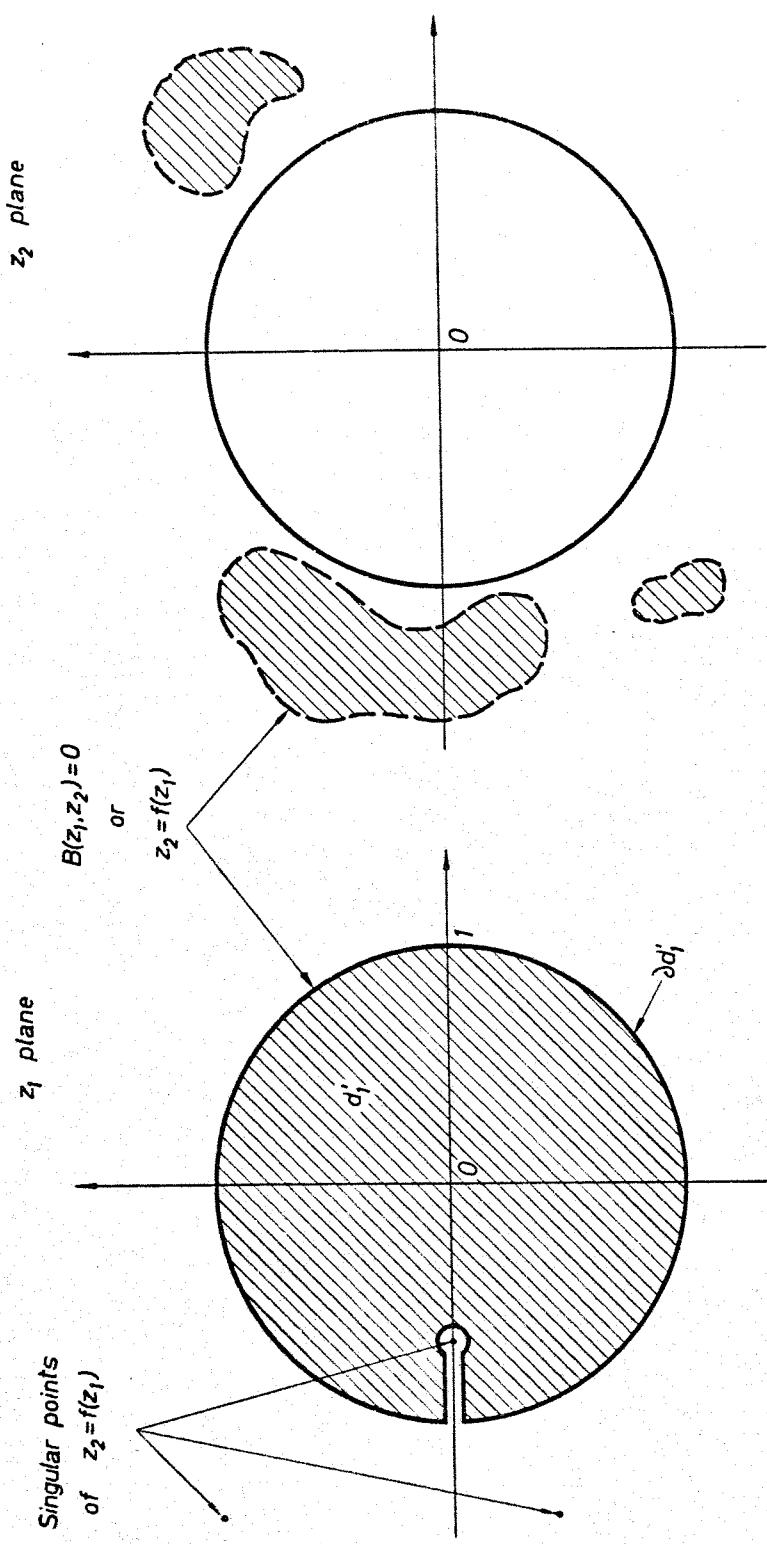


Fig. A.1. Mapping $z_2 = f(z_1)$.

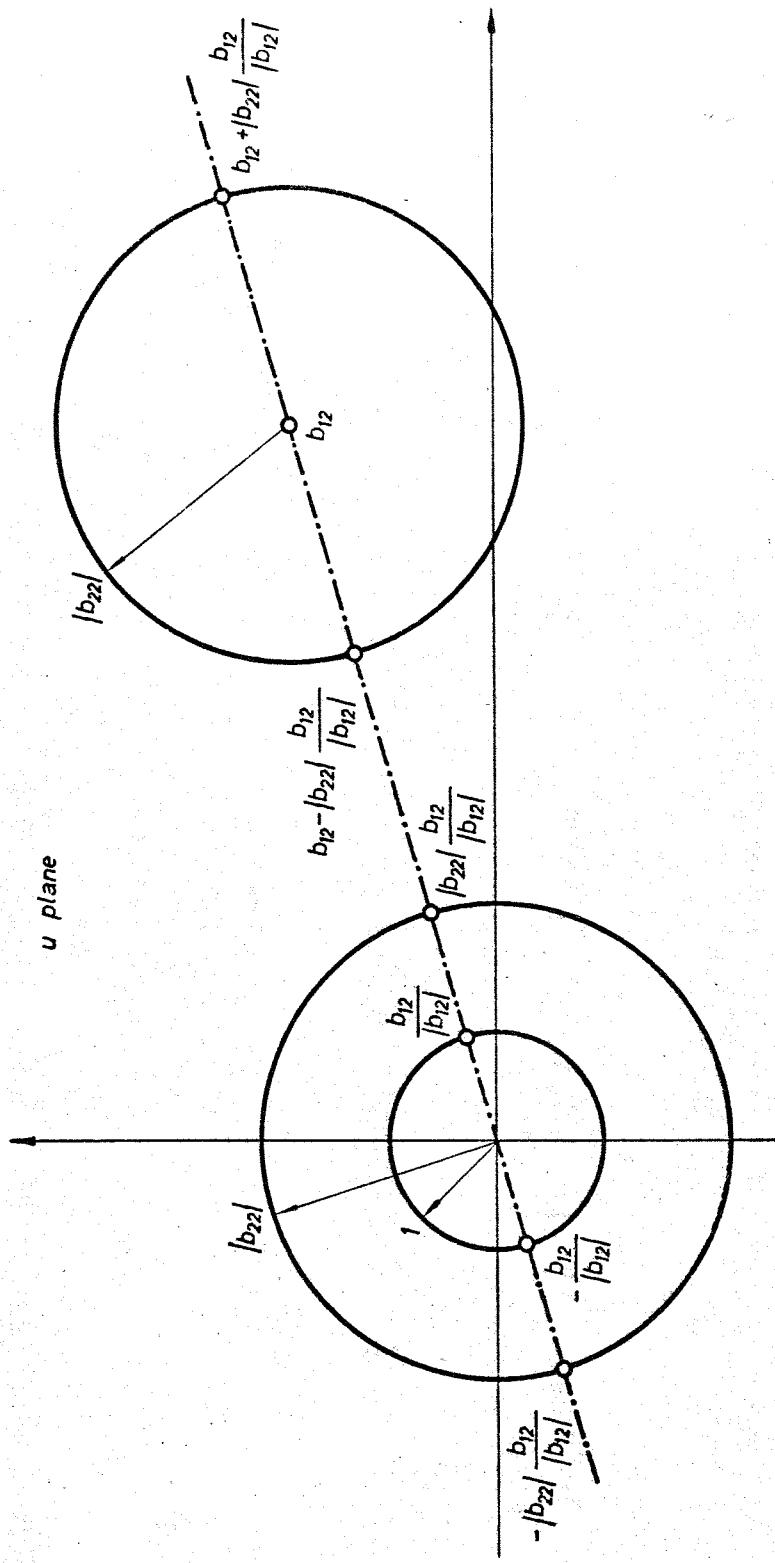


Fig. C.1. Mapping the unit circle by the transformation $u = b_{12} + b_{22} z_1$.

