

Stability Criterion for Recursive Filters

Abstract: A new criterion is derived that relates the stability of two-dimensional recursive filters to the properties of its cepstrum. It provides a procedure for the decomposition of unstable recursive filters having nonzero, nonimaginary frequency response into stable recursive filters. The optimal solution of the decomposition problem is discussed, including numerical implementation and nonrecursive solutions. Several numerical examples show the potentialities and limitations of the rules for decomposition and for truncation of the operators.

Contents

Introduction

Space domain operations and their z transforms

z transform of two-dimensional functions

Convolutional filters

Recursive filters

Two-dimensional cepstrum

Decomposition of unstable, recursive, zero-phase filters into stable recursive filters

Stability criterion for recursive filters based on two-dimensional cepstra

Solution of decomposition problem

Existence of a set of solutions

Problem of an optimal solution

Truncation problem

Numerical implementation

Nonrecursive solutions

Numerical examples

Example 1

Example 2

Example 3

Concluding remarks

Acknowledgments

References

Introduction

In the development of two-dimensional recursive filters the two problem areas of main interest are the stability criteria and the design of stable recursive filters. Except for filters with very few samples, existing stability criteria [1-4] are difficult to apply and thus it has been virtually impossible to translate these criteria into a practical design procedure. Moreover, the criteria could not be used to justify a design procedure [2] that appears to provide stable recursive filters.

The present paper proposes a criterion that is based on the relationship of the stability of recursive filters to the absolute summability of certain operators called *cepstra* [5]. The z transform of these operators is given by the logarithm of the z transform of the recursive filters. By means of this criterion we treat a special design problem [4] concerning the decomposition of unstable recursive filters having nonzero and nonimaginary frequency response into stable recursive filters. For the case of one-dimensional filters this problem was solved by the decomposition into minimum-delay and maximum-delay operators [6, 7]. Our stability criterion leads to a proof of the existence of a similar solution for the two-dimensional case.

This paper is organized into five main sections: a compilation, for reference purposes, of z transform relations; a demonstration of the need for a decomposition procedure for zero-phase filters; a proof of the stability criterion; a solution of the decomposition problem, including methods of numerical implementation and related nonrecursive solutions; and some numerical examples.

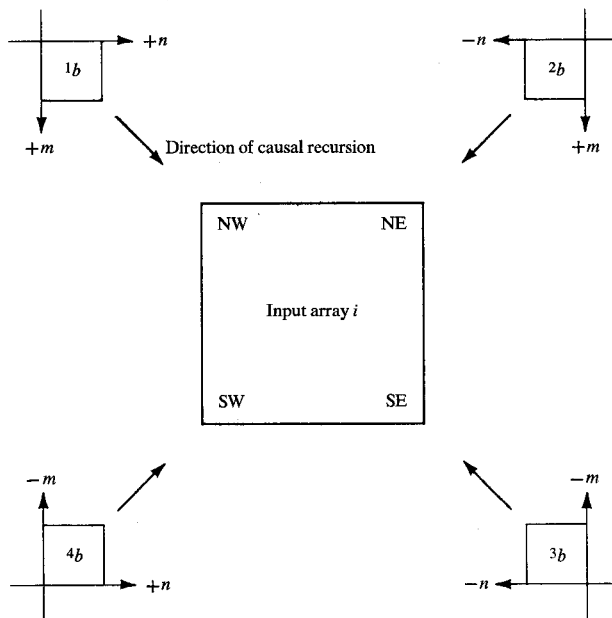


Figure 1 Four directions of recursion from the four corners of an input array. The input array is to be understood as a first-, second-, third-, or fourth-quadrant function, depending on the direction of the recursion.

Space domain operations and their z transforms

The z transform is a convenient means for characterizing certain operations involving sampled functions. Some relations [1-4, 8] needed for subsequent discussions are summarized in this section.

• Z transform of two-dimensional functions

The z transform of a periodically sampled, two-dimensional function

$$b = \{b_{m,n}\} \quad (1)$$

is defined by

$$B(z_1, z_2) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} b_{m,n} z_1^m z_2^n, \quad (2)$$

where z_1 and z_2 denote complex variables, and $b_{m,n}$ is understood to be zero for subscript pairs (m,n) that do not belong to the definition set of b .

If b is defined for nonnegative subscripts only, it is called a *first-quadrant function* [2] and is denoted by a left superscript as

$$^1b = \{^1b_{m,n}\}_{\substack{m \geq 0 \\ n \geq 0}} \text{ (first-quadrant function).} \quad (3)$$

Similarly, second-, third- and fourth-quadrant functions can be defined successively as

$$^2b = \{^2b_{m,n}\}_{\substack{m \geq 0 \\ n \leq 0}} \quad (4)$$

$$^3b = \{^3b_{m,n}\}_{\substack{m \leq 0 \\ n \geq 0}} \quad (5)$$

$$^4b = \{^4b_{m,n}\}_{\substack{m \leq 0 \\ n \leq 0}} \text{ (see [9]).} \quad (6)$$

The z transform pairs in Eqs. (7) - (12), denoted by $\leftrightarrow$, relate some basic spatial operations to operations in the z transform domain [2, 8]. These relations can readily be improved by comparing terms of the same degree in z_1 and z_2 .

Basic pair:

$$\{b_{m,n}\} \leftrightarrow B(z_1, z_2)$$

Translation:

$$\{b_{m,n}'\} = \{b_{m-\alpha, n-\beta}\} \leftrightarrow B'(z_1, z_2) = z_1^\alpha z_2^\beta B(z_1, z_2), \quad (7)$$

Rotation through 180° around $(0,0)$:

$$\{b_{m,n}'\} = \{b_{-m, -n}\} \leftrightarrow B'(z_1, z_2) = B(1/z_1, 1/z_2), \quad (8)$$

Reflection at axis $m = 0$ (column reversion):

$$\{b_{m,n}'\} = \{b_{-m, n}\} \leftrightarrow B'(z_1, z_2) = B(1/z_1, z_2), \quad (9)$$

Reflection at axis $n = 0$ (row reversion):

$$\{b_{m,n}'\} = \{b_{m, -n}\} \leftrightarrow B'(z_1, z_2) = B(z_1, 1/z_2), \quad (10)$$

Weighting by row number:

$$\{b_{m,n}'\} = \{mb_{m,n}\} \leftrightarrow B'(z_1, z_2) = z_1 (\partial / \partial z_1) B(z_1, z_2), \quad (11)$$

Weighting by column number:

$$\{b_{m,n}'\} = \{nb_{m,n}\} \leftrightarrow B'(z_1, z_2) = z_2 (\partial / \partial z_2) B(z_1, z_2). \quad (12)$$

• Convolutional filters

The input-output relation of a convolutional filter f ,

$$o = i * f, \quad (13)$$

where

$$o_{p,q} = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} i_{m,n} f_{p-m, q-n}, \quad (14)$$

is written in z transform notation as the product

$$O(z_1, z_2) = I(z_1, z_2) F(z_1, z_2). \quad (15)$$

A formal proof of this equation may be found by the same method as for Eqs. (7) - (12).

Definition 1

A filter is stable if and only if its response to any bounded input is bounded.

For a convolutional filter the stability criterion [3,4], Theorem 1, can be proved.

Theorem 1

A convolutional filter is stable if and only if

$$\sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} |f_{m,n}| < \infty. \quad (16)$$

Condition (16) means that the z transform of f is absolutely convergent for all $(z_1, z_2) \in R$,

$$R = \{(z_1, z_2) : |z_1| = 1 \wedge |z_2| = 1\}. \quad (17)$$

• Recursive filters

The convolution of two discrete functions corresponds to a multiplication of their z transforms, whereas the input-output relation of recursive filters is characterized in the z transform domain by the quotient [10]

$$O(z_1, z_2) = I(z_1, z_2) / B(z_1, z_2) = I(z_1, z_2) F(z_1, z_2). \quad (18)$$

If O , I , and B are replaced by first-quadrant functions 1o , 1i , and 1b with

$${}^1i = \{i_{m,n}\}_{\substack{0 \leq m \leq M_i \\ 0 \leq n \leq N_i}}, \quad (19)$$

$${}^1b = \{b_{m,n}\}_{\substack{0 \leq m \leq M_b \\ 0 \leq n \leq N_b}}, \quad (20)$$

the space domain algorithm corresponding to Eq. (18) is

$${}^1o_{0,0} = {}^1i_{0,0} / {}^1b_{0,0},$$

$${}^1o_{p,q} = (1 / {}^1b_{0,0}) \left(i_{p,q} - \sum_{m=0}^{\min(p, M_b)} \sum_{n=0}^{\min(q, N_b)} {}^1b_{m,n} {}^1o_{p-m, q-n} \right),$$

$$(m, n) \neq (0, 0)$$

$$(p, q) \in \{(p, q) : (p, q) \neq (0, 0) \wedge p \leq M_i \wedge q \leq N_i\}$$

$${}^1o_{p,q} = (1 / {}^1b_{0,0}) \left(- \sum_{m=0}^{\min(p, M_b)} \sum_{n=0}^{\min(q, N_b)} {}^1b_{m,n} {}^1o_{p-m, q-n} \right),$$

$$(m, n) \neq (0, 0).$$

$$(p, q) \in \{(p, q) : (p, q) \neq (0, 0) \wedge (p > M_i \vee q > N_i)\}. \quad (21)$$

These formulas strongly resemble the convolution algorithm. In fact, they can be derived from Eq. (14) by replacing o , i , and f by 1i , 1b , and 1o , respectively, and solving for ${}^1o_{p,q}$.

To illustrate the use of Eqs. (21), consider the input

$${}^1i = \begin{bmatrix} {}^1i_{0,0} & {}^1i_{0,1} & {}^1i_{0,2} \\ {}^1i_{1,0} & {}^1i_{1,1} & {}^1i_{1,2} \\ {}^1i_{2,0} & {}^1i_{2,1} & {}^1i_{2,2} \end{bmatrix} = \begin{bmatrix} 3 & 4 & 1 \\ 4 & 5 & 1 \\ 1 & 1 & 0 \end{bmatrix},$$

which is to be recursively filtered by

$${}^1b = \begin{bmatrix} {}^1b_{0,0} & {}^1b_{0,1} \\ {}^1b_{1,0} & {}^1b_{1,1} \end{bmatrix} = \begin{bmatrix} 3 & 1 \\ 1 & 0 \end{bmatrix}.$$

The output value to be determined first is ${}^1o_{0,0}$:

$${}^1o_{0,0} = {}^1i_{0,0} / {}^1b_{0,0} = 3/3 = 1.$$

To understand the operation of the last two formulas of Eqs. (21), we form the mask from 1b ,

$$\begin{bmatrix} 0 & 1 \\ 1 & 3 \end{bmatrix},$$

and draw it across the matrices 1i and 1o synchronously. In the output arrays 1o of the following examples, the symbol * designates elements not yet determined. The numeral 3 refers to the value just computed. The other entries of the mask form a dot product with the output values to which they refer. These values must be known, of course, from former steps. The result of the dot product is subtracted from the input value indicated by 3, and this result is then divided by ${}^1b_{0,0}$.

The examples that follow demonstrate the computation of ${}^1o_{1,0}$, ${}^1o_{2,0}$, ${}^1o_{3,0}$ and ${}^1o_{1,1}$.

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 3 & 4 & 1 \\ 0 & 4 & 5 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & * \\ 0 & 1 & 3 & * \\ 0 & * & * & * \end{bmatrix}, \quad {}^1o_{1,0} = \frac{1}{3}(4 - 1 \cdot 1 - 1 \cdot 0 - 0 \cdot 0) = 1;$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 3 & 4 & 1 \\ 0 & 4 & 5 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & * \\ 0 & 1 & 3 & * \\ 0 & * & * & * \end{bmatrix}, \quad {}^1o_{2,0} = \frac{1}{3}(1 - 1 \cdot 1 - 1 \cdot 0 - 0 \cdot 0) = 0;$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 3 & 4 & 1 \\ 0 & 4 & 5 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & * \\ 0 & 1 & 3 & * \\ 0 & * & * & * \end{bmatrix}, \quad {}^1o_{3,0} = \frac{1}{3}(0 - 1 \cdot 0 - 1 \cdot 0 - 0 \cdot 0) = 0;$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 3 & 4 & 1 \\ 0 & 4 & 5 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & * \\ 0 & 1 & 3 & * \\ 0 & * & * & * \end{bmatrix}, \quad {}^1o_{1,1} = \frac{1}{3}(5 - 1 \cdot 1 - 1 \cdot 1 - 1 \cdot 0) = 1.$$

Equations (21) describe the so-called *causal recursion* [4], in which the recursive operator 1b starts from the NW corner of an input array, as indicated in Fig. 1. Similar algorithms can be derived for functions i , o , and b defined on the other quadrants. The equations describe *noncausal recursions*, i.e., recursions that start from the NE, SE, and SW corners of an input array. Any noncausal recursion, however, can be transformed to the causal recursion. For instance, the noncausal recursive filtering of an input 3i by an operator 3b is transformed to a recursion with an operator 1b simply by rotation of functions 3i and 3b through 180° . Because of Eq. (8), such rotations correspond to a replacement of (z_1, z_2) by $(1/z_1, 1/z_2)$. The transformation of 3i and 3b thus corresponds to the transition from

$$O_3(z_1, z_2) = I_3(z_1, z_2) / B_3(z_1, z_2) \quad (22)$$

to

$$O_3(1/z_1, 1/z_2) = I_3(1/z_1, 1/z_2) / B_3(1/z_1, 1/z_2) \quad (23)$$

in the z transform domain. This input-output relation is equivalent to

$$O_1(z_1, z_2) = I_1(z_1, z_2) / B_1(z_1, z_2). \quad (24)$$

For illustration, the input

$${}^3i = \begin{bmatrix} {}^3i_{-2,-2} & {}^3i_{-2,-1} & {}^3i_{-2,0} \\ {}^3i_{-1,-2} & {}^3i_{-1,-1} & {}^3i_{-1,0} \\ {}^3i_{0,-2} & {}^3i_{0,-1} & {}^3i_{0,0} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 5 & 4 \\ 1 & 4 & 3 \end{bmatrix}$$

is recursively filtered by

$${}^3b = \begin{bmatrix} {}^3b_{-1,-1} & {}^3b_{-1,0} \\ {}^3b_{0,-1} & {}^3b_{0,0} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 3 \end{bmatrix}.$$

Instead of performing the appropriate noncausal recursion, which would start from the lower-right input element ${}^3i_{0,0}$, we rotate 3i and 3b , obtaining the functions 1i and 1b discussed in the preceding example. From the intermediate result,

$${}^1o = \begin{bmatrix} {}^1o_{0,0} & {}^1o_{0,1} & {}^1o_{0,2} & {}^1o_{0,3} & \dots \\ {}^1o_{1,0} & {}^1o_{1,1} & {}^1o_{1,2} & {}^1o_{1,3} & \dots \\ {}^1o_{2,0} & {}^1o_{2,1} & {}^1o_{2,2} & {}^1o_{2,3} & \dots \\ {}^1o_{3,0} & {}^1o_{3,1} & {}^1o_{3,2} & {}^1o_{3,3} & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 & \dots \\ 1 & 1 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix},$$

we obtain the final result by inverting the above rotations as

$${}^3o = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & 0 & 0 & 0 \\ \vdots & \vdots & 0 & 1 & 1 \\ \vdots & \vdots & 0 & 1 & 1 \end{bmatrix}.$$

Similarly, the replacement of z_1 by $1/z_2$, or z_2 by $1/z_1$, respectively, corresponds to the transition of the other noncausal recursions to the causal recursion.

If we put

$${}^1i_{m,n} = \delta_{m,n} = \begin{cases} 1, & m = n = 0; \\ 0, & \text{else,} \end{cases} \quad (25)$$

the recursive algorithm (21) yields the convolutional filter 1f , equivalent to the recursive filter $1/B_1(z_1, z_2) = F_1(z_1, z_2)$. According to theorem 1, the stability of these filters may be related as indicated in theorems 2, 3, and the subsequent discussion.

Theorem 2

A causal recursive filter $1/B_1(z_1, z_2)$ is stable if and only if there exists a stable filter 1f such that ${}^1f * {}^1b = \delta$.

Farmer and Bednar [3] have given a stability criterion for recursive filters. This criterion comprises that of Shanks [2], which is confined to operators 1b with a finite number of sample points.

Theorem 3

The causal recursive filter

$$F(z_1, z_2) = 1/B_1(z_1, z_2) = 1 / \left(\sum_{m=0}^{\infty} \sum_{n=0}^{\infty} {}^1b_{m,n} z_1^m z_2^n \right), \quad (26)$$

where

$$\sum_{m=0}^{\infty} \sum_{n=0}^{\infty} |{}^1b_{m,n}| < \infty \quad (27)$$

is stable if and only if

$$B_1(z_1, z_2) \neq 0 \quad (28)$$

for all $(z_1, z_2) \in D_1$, where

$$D_1 = \{(z_1, z_2): |z_1| \leq 1 \wedge |z_2| \leq 1\}. \quad (29)$$

The foregoing discussion shows how noncausal recursions can be transformed to the causal recursion. The following corollary is thus evident.

Corollary 1

Let the l th-quadrant function ${}^l b$ ($l = 2, 3, 4$) be a stable convolutional filter. Then the noncausal recursive filters $1/B_l(z_1, z_2)$ are stable if and only if $B_2(z_1, 1/z_2)$, $B_3(1/z_1, 1/z_2)$, and $B_4(1/z_1, z_2)$ are nonzero for all $(z_1, z_2) \in D_1$.

The functions that meet the conditions of Theorem 3 and Corollary 1 are characterized by Definition 2.

Definition 2

A one-quadrant function ${}^l b$ is called recursively stable if and only if ${}^l b$ is a stable convolutional filter and $1/B_l(z_1, z_2)$ is a stable recursive filter, either causal or noncausal.

• Two-dimensional cepstrum

We consider once more a function b and its z transform:

$$\{b_{m,n}\} \leftrightarrow B(z_1, z_2). \quad (30)$$

By formally taking the logarithm of B we obtain

$$\hat{B}(z_1, z_2) = \ln B(z_1, z_2). \quad (30')$$

The cepstrum of b , i.e., $\hat{b}$, is a function whose z transform is given by $\hat{B}(z_1, z_2)$:

$$\{\hat{b}_{m,n}\} \leftrightarrow \hat{B}(z_1, z_2) = \ln B(z_1, z_2). \quad (30'')$$

According to Eqs. (30), (30'), (30''), (13), and (15), we find, for the input-output relation $o = i * f$ of a convolutional filter,

$$\hat{O}(z_1, z_2) = \hat{I}(z_1, z_2) + \hat{F}(z_1, z_2),$$

which in turn leads to the additive relation in the cepstrum domain,

$$\hat{o} = \hat{i} + \hat{f}. \quad (31)$$

Similarly, the input-output relation (18) of a recursive filter is given by a difference in the cepstrum domain,

$$\hat{o} = \hat{i} - \hat{b}. \quad (32)$$

The problem stated in the next section is solved by means of cepstra.

Decomposition of unstable, recursive, zero-phase filters into stable recursive filters

We consider a real-valued discrete function c with a limited number of sample points, in which

$$c = \{c_{m,n}\}_{\substack{|m| \leq 2M_c = \alpha \\ |n| \leq 2N_c = \beta}} \quad (33)$$

Its z transform is of zero phase and non-negative for all $(z_1, z_2) \in R$:

$$\text{Im}[C'(u, v)] = 0, \quad (34)$$

$$C'(u, v) > 0, \quad (35)$$

where

$$C'(u, v) = \sum_{m=-\alpha}^{\alpha} \sum_{n=-\beta}^{\beta} c_{m,n} e^{-2\pi j u m} e^{-2\pi j v n} \quad j = \sqrt{-1} \\ = C(z_1, z_2)_{(z_1, z_2) \in R}. \quad (36)$$

Equations (34) and (36) imply central symmetry of c :

$$c_{m,n} = c_{-m,-n}. \quad (37)$$

Functions as specified in Eq. (33)ff. are found with recursive filters $F(z_1, z_2) = A(z_1, z_2)/C(z_1, z_2)$ [11] for the enhancement of images degraded by noise and by linear shift-invariant geometric distortion.

Because of Eq. (37), c is not a one-quadrant function. However, because c has a limited number of sample points, it can be transformed by translation to any quadrant function. Thus the term $1/C(z_1, z_2)$ could be associated with four different recursive filters. None, however, would be stable.

Proof By translation, the following z transforms originate from c [see Eq. (7)]:

$$C_1(z_1, z_2) = z_1^\alpha z_2^\beta C(z_1, z_2), \quad \alpha = 2M_c, \beta = 2N_c \quad (38)$$

$$C_2(z_1, z_2) = z_1^\alpha z_2^{-\beta} C(z_1, z_2), \quad (39)$$

$$C_3(z_1, z_2) = z_1^{-\alpha} z_2^{-\beta} C(z_1, z_2), \quad (40)$$

$$C_4(z_1, z_2) = z_1^{-\alpha} z_2^\beta C(z_1, z_2). \quad (41)$$

According to Theorem 3 and Corollary 1, the recursive filters associated with these z transforms are unstable if a

pair $(z_1, z_2) \in D_1$ exists such that

$$C_1(z_1, z_2) = C_2(z_1, 1/z_2) \\ = C_3(1/z_1, 1/z_2) = C_4(1/z_1, z_2) = 0. \quad (42)$$

By defining $z_2 = 1$, the existence of such a pair is proved, if a $z_1 = w$, $|w| \leq 1$, can be found, which solves simultaneously the equations

$$z_1^\alpha C(z_1, 1) = \sum_{m=-\alpha}^{\alpha} \sum_{n=-\beta}^{\beta} c_{m,n} z_1^{\alpha+m} = z_1^\alpha G(z_1) = 0, \quad (43)$$

$$z_1^\alpha C(1/z_1, 1) = \sum_{m=-\alpha}^{\alpha} \sum_{n=-\beta}^{\beta} c_{m,n} z_1^{\alpha-m} \\ = z_1^\alpha G(1/z_1) = 0. \quad (44)$$

Since it can be concluded from the central symmetry of c that

$$G(z_1) = \sum_{m=-\alpha}^{\alpha} \sum_{n=-\beta}^{\beta} c_{m,n} z_1^m = G(1/z_1), \quad (45)$$

Eqs. (43) and (44) are identical. If they are solved by $z_1 = 0$, the sought pair is $(0, 1)$. If not, Eq. (45) shows that every root $z_1 = w$ has a corresponding root $z_1 = 1/w$. Thus a pair $(z_1, z_2) \in D_1$ can always be found, solving Eqs. (42), and therefore no recursively stable operator can be obtained from c by translation.

In the one-dimensional case, unstable recursive filters $1/B(z_1)$, with no poles on the unit circle, can be decomposed into stable filters that recurse in opposite directions. Therefore the question arises whether, analogously, the unstable filter $1/C(z_1, z_2)$ can be decomposed into four stable filters that recurse in four different directions. More precisely, we ask for recursively stable one-quadrant functions 1k , 2k , 3k , and 4k such that

$$1/C(z_1, z_2) = \prod_{l=1}^4 1/K_l^{-1}(z_1, z_2). \quad (46)$$

In a way similar to the solution for the one-dimensional case [12], we find a solution to this problem by transforming Eq. (46) to the cepstrum domain [viz., Eq. (30)],

$$\hat{c} = \sum_{l=1}^4 \hat{k}_l. \quad (47)$$

The results of the next section indicate how to decompose $\hat{c}$ and how to use the resulting functions to compute the operators that are sought.

Stability criterion for recursive filters based on two-dimensional cepstra

Let us assume that 1b is a first-quadrant function that is recursively stable. This assumption implies, from Theorems 1, 2, and 3 and Definition 2, that

$$\sum_{m=0}^{\infty} \sum_{n=0}^{\infty} |^1f_{m,n}| < \infty, \quad (48)$$

$${}^1b * {}^1f = \{\delta_{m,n}\}, \quad (49)$$

$$B_1(z_1, z_2) \neq 0, (z_1, z_2) \in D_1, \quad (50)$$

$$\sum_{m=0}^{\infty} \sum_{n=0}^{\infty} |{}^1b_{m,n}| < \infty. \quad (51)$$

From these facts we conclude that $\hat{B}_1(z_1, z_2) = \ln B_1(z_1, z_2)$ can be uniquely expanded into the power series

$$\sum_{m=0}^{\infty} \sum_{n=0}^{\infty} {}^1\hat{b}_{m,n} z_1^m z_2^n \quad (52)$$

for all $(z_1, z_2) \in D_1$.

First we consider the case $(z_1, z_2) \in D_2$,

$$D_2 = \{(z_1, z_2) : |z_1| < 1 \wedge |z_2| < 1\}. \quad (53)$$

On this set $B_1(z_1, z_2)$ is regular [13], due to Eq. (51), and consequently $\hat{B}_1(z_1, z_2)$ is regular, too, because $B_1(z_1, z_2)$ is nonzero [condition (50)]. Thus a unique expansion (52) of $\hat{B}_1$ exists. The coefficient ${}^1\hat{b}_{0,0}$ of this power series is readily determined by evaluating

$$\hat{B}_1(z_1, z_2) = \ln B_1(z_1, z_2) \quad (54)$$

for $(z_1, z_2) = (0, 0)$:

$${}^1\hat{b}_{0,0} = \ln {}^1b_{0,0}. \quad (55)$$

To find the other coefficients we differentiate Eq. (54) with respect to z_1 and z_2 and replace $1/B_1(z_1, z_2)$ by $F_1(z_1, z_2)$:

$$z_1 (\partial / \partial z_1) \hat{B}_1(z_1, z_2) = F_1(z_1, z_2) z_1 (\partial / \partial z_1) B_1(z_1, z_2), \quad (56)$$

$$z_2 (\partial / \partial z_2) \hat{B}_1(z_1, z_2) = F_1(z_1, z_2) z_2 (\partial / \partial z_2) B_1(z_1, z_2). \quad (57)$$

According to Eqs. (11), (12), (13), and (15) these relations correspond to

$$\{m {}^1\hat{b}_{m,n}\} = \{{}^1f_{m,n}\} * \{m {}^1b_{m,n}\} \quad (58)$$

and

$$\{n {}^1\hat{b}_{m,n}\} = \{{}^1f_{m,n}\} * \{n {}^1b_{m,n}\}, \quad (59)$$

or, explicitly written,

$${}^1\hat{b}_{p,q} = \sum_{m=0}^p \sum_{n=0}^q (m/p) {}^1b_{m,n} {}^1f_{p-m, q-n}, \quad p \neq 0 \quad (60)$$

$${}^1\hat{b}_{p,q} = \sum_{m=0}^p \sum_{n=0}^q (n/q) {}^1b_{m,n} {}^1f_{p-m, q-n}, \quad q \neq 0. \quad (61)$$

We prove now that the power series defined by Eqs. (52), (55), (60), and (61) is absolutely convergent and equal to $\hat{B}_1(z_1, z_2)$ for all $(z_1, z_2) \in D_1 - D_2$. From Eqs. (60) and (61) we find that, for all $(z_1, z_2) \in D_1$,

$$\sum_{p=0}^N \sum_{q=0}^N |{}^1\hat{b}_{p,q}| |z_1|^p |z_2|^q \leq \sum_{q=1}^N \sum_{n=0}^q (n/q) |{}^1b_{0,n}| |{}^1f_{0,q-n}|$$

$$(p, q) \neq (0, 0),$$

$$+ \sum_{p=1}^N \sum_{q=0}^N \sum_{m=0}^p \sum_{n=0}^q (m/p) |{}^1b_{m,n}| |{}^1f_{p-m, q-n}|$$

$$\leq \sum_{p=0}^N \sum_{q=0}^N \sum_{m=0}^p \sum_{n=0}^q |{}^1b_{m,n}| |{}^1f_{p-m, q-n}|$$

$$\leq \sum_{m=0}^N \sum_{n=0}^N |{}^1b_{m,n}| \sum_{p=0}^N \sum_{q=0}^N |{}^1f_{p,q}|,$$

and thus

$$\sum_{p=0}^{\infty} \sum_{q=0}^{\infty} |{}^1\hat{b}_{p,q}| |z_1|^p |z_2|^q \leq \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} |{}^1b_{m,n}| \sum_{p=0}^{\infty} \sum_{q=0}^{\infty} |{}^1f_{p,q}| < \infty;$$

$$(p, q) \neq (0, 0), (z_1, z_2) \in D_1.$$

(for left-hand part only). (62)

In Eq. (62) we have used the stability of 1f [Eq. (48)] and 1b [Eq. (51)].

With the intermediate notation

$$E(z_1, z_2) = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} {}^1\hat{b}_{m,n} z_1^m z_2^n, \quad (63)$$

it remains to be proved that

$$E(z_1, z_2) = \hat{B}_1(z_1, z_2)$$

for all $(z_1, z_2) \in D_1 - D_2$. Because $E(z_1, z_2)$ and $B_1(z_1, z_2)$ are given by power series that are absolutely convergent for all $(z_1, z_2) \in D_1$, it follows from Abel's second theorem [14] that

$$\lim_{\rho \rightarrow 1-0} [B_1(e^{-2\pi ju}, z_2) - B_1(\rho e^{-2\pi ju}, z_2)] = 0 \quad (64)$$

and

$$\lim_{\rho \rightarrow 1-0} [E(e^{-2\pi ju}, z_2) - E(\rho e^{-2\pi ju}, z_2)] = 0 \quad (65)$$

for any fixed (u, z_2) with

$$(u, z_2) \in \{(u, z_2) : 0 \leq u \leq 1 \wedge |z_2| < 1\} = Q.$$

From (64) we conclude [referring to condition (50) and the continuity of the logarithm] that

$$\lim_{\rho \rightarrow 1-0} [\hat{B}_1(e^{-2\pi ju}, z_2) - \hat{B}_1(\rho e^{-2\pi ju}, z_2)] = 0 \quad (66)$$

for any $(u, z_2) \in Q$. When (66) is subtracted from (65),

$$\hat{B}_1(z_1, z_2) = E(z_1, z_2) \quad (67)$$

for all $(z_1, z_2) \in \{(z_1, z_2) : |z_1| \leq 1 \wedge |z_2| < 1\}$.

Repeating the argument for $z_1 = \exp(-2\pi ju)$ and $z_2 = \rho \exp(-2\pi jv)$, where u and v are any fixed values, we finally find

$$\hat{B}_1(z_1, z_2) = E(z_1, z_2), (z_1, z_2) \in D_1. \quad (68)$$

Thus the convolutional stability of ${}^1\hat{b}$ is a necessary condition for 1b to be recursively stable. We prove now that this property is also a sufficient condition. Again, we first consider the set D_2 . On D_2 , $\hat{B}_1(z_1, z_2)$ is regular because of Eq. (62), and because $B_1(z_1, z_2) = \exp[\hat{B}_1(z_1, z_2)]$ is regular, too, the identities

$$z_1(\partial/\partial z_1) B_1(z_1, z_2) = B_1(z_1, z_2) z_1(\partial/\partial z_1) \hat{B}_1(z_1, z_2) \quad (69)$$

and

$$z_2(\partial/\partial z_2) B_1(z_1, z_2) = B_1(z_1, z_2) z_2(\partial/\partial z_2) \hat{B}_1(z_1, z_2) \quad (70)$$

relate the absolutely convergent power series. These identities yield relations similar to Eqs. (60) and (61):

$${}^1b_{p,q} = \sum_{m=1}^p \sum_{n=0}^q (m/p) {}^1\hat{b}_{m,n} {}^1b_{p-m,q-n}, \quad p \neq 0, \quad (71)$$

$${}^1b_{p,q} = \sum_{m=0}^p \sum_{n=1}^q (n/q) {}^1\hat{b}_{m,n} {}^1b_{p-m,q-n}, \quad q \neq 0. \quad (72)$$

For $(p, q) = (0, 0)$ we find, from Eq. (55),

$${}^1b_{0,0} = \exp({}^1\hat{b}_{0,0}). \quad (73)$$

With 1b as defined by Eqs. (71)–(73), we have the

$$B_1(z_1, z_2) = \exp[\hat{B}_1(z_1, z_2)] = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} {}^1b_{m,n} z_1^m z_2^n \quad (74)$$

identity for all $(z_1, z_2) \in D_2$. We prove now that (74) is even valid for all $(z_1, z_2) \in D_1$. To do this it is sufficient to find a power series that is equal to $B_1(z_1, z_2)$ and absolutely convergent for all $(z_1, z_2) \in D_1$.

Since $\hat{B}_1(z_1, z_2)$ is given as a power series that is, by assumption, absolutely convergent for all $(z_1, z_2) \in D_1$, and the expansion of $\exp(x)$ is absolutely convergent for all $|x| < \infty$, the expansion

$$\begin{aligned} \exp[\hat{B}_1(z_1, z_2)] &= \sum_{l=0}^{\infty} \frac{1}{l!} [\hat{B}_1(z_1, z_2)]^l \\ &= \sum_{l=0}^{\infty} (1/l!) \left(\sum_{m=0}^{\infty} \sum_{n=0}^{\infty} {}^1\hat{b}_{m,n} z_1^m z_2^n \right)^l \end{aligned} \quad (75)$$

is also absolutely convergent for all $(z_1, z_2) \in D_1$. Thus Eq. (75) can be rearranged as

$$\exp[\hat{B}_1(z_1, z_2)] = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} a_{m,n} z_1^m z_2^n, \quad (76)$$

where

$$a_{m,n} = \sum_{l=0}^{\infty} a_{m,n}^{(l)}, \quad (77)$$

$$\begin{aligned} a_{m,n}^{(l)} &= (1/l) \sum_{p=0}^m \sum_{q=0}^n a_{p,q}^{(l-1)} {}^1\hat{b}_{m-p,n-q}, \\ l &= 1, 2, \dots, \end{aligned} \quad (78)$$

$$a_{m,n}^{(0)} = \delta_{m,n}. \quad (79)$$

Therefore the expansion (74) is valid and absolutely convergent for all $(z_1, z_2) \in D_1$. It is evident from the definition that $B_1(z_1, z_2)$ is nonzero on D_1 . Theorem 4 summarizes the previous results.

Theorem 4. The quantity

$$\{ {}^1b_{m,n} \}_{\substack{m \geq 0 \\ n \geq 0}} \quad (80)$$

is recursively stable if and only if there exists a power series

$$\sum_{m=0}^{\infty} \sum_{n=0}^{\infty} {}^1\hat{b}_{m,n} z_1^m z_2^n$$

that is absolutely convergent and equal to $\ln B_1(z_1, z_2)$ for all $(z_1, z_2) \in D_1$.

Theorem 4 remains applicable to noncausal recursive operators ${}^l b$, $l = 2, 3, 4$, if they are transformed to a first-quadrant function by the operations (10), (8), or (9), respectively. This leads to Corollary 2.

Corollary 2

The l th-quadrant function ${}^l b$, in which $l = 2, 3, 4$, is recursively stable if and only if $\ln [B_2(z_1, 1/z_2)]$, $\ln [B_3(1/z_1, 1/z_2)]$, or $\ln [B_4(1/z_1, z_2)]$, respectively, is equal to a power series of the form (80) that is absolutely convergent for all $(z_1, z_2) \in D_1$.

From Theorem 4 and Corollary 2 it is evident that the decomposition problem in the previous section can be solved if the cepstrum $\hat{c}$ of c can be decomposed [see Eq. (47)] into one-quadrant functions ${}^l \hat{k}$, in which $l = 1, 2, 3, 4$. These are stable convolutional filters. In the next sections it is proved that the decomposition is possible because of the properties of c .

Solution of the decomposition problem

Existence of a set of solutions

The cepstrum $\hat{c}$ is readily obtained by evaluating the z -transform of c on R . We first have

$$\hat{C}'(u, v) = \ln \left[\sum_{m=-2M_c}^{2M_c} \sum_{n=-2N_c}^{2N_c} c_{m,n} e^{-2\pi j(um+vn)} \right]. \quad (81)$$

Since $\hat{C}'$ is real and positive [Eqs. (34) and (35)], the logarithm in Eq. (81) presents no problem. It is easily verified that all partial derivatives of $\hat{C}'(u, v)$ are continuous. Thus, for all u and v , $\hat{C}'(u, v)$ is given by the Fourier series expansion [15]

$$\hat{C}'(u, v) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \hat{c}_{m,n} e^{-2\pi j(um+vn)}. \quad (82)$$

The coefficients $\hat{c}_{m,n}$ that make up the cepstrum $\hat{c}$ are obtained from

$$\hat{c}_{m,n} = \int_0^1 \int_0^1 \hat{C}'(u, v) e^{2\pi j(um+vn)} dudv. \quad (83)$$

From the properties of $\hat{C}'(u, v)$ and from Eq. (83) it follows that

$$\hat{c}_{m,n} = \hat{c}_{-m,-n}. \quad (84)$$

Repeated integration by parts permits proof of the inequalities

$$|\hat{c}_{0,0}| \leq \alpha < \infty, \quad (85)$$

$$|\hat{c}_{0,n}| \leq \alpha/n^2, \quad n \neq 0, \quad (86)$$

$$|\hat{c}_{m,0}| \leq \alpha/m^2, \quad m \neq 0, \quad (87)$$

$$|\hat{c}_{m,n}| \leq \alpha/(m^2 n^2), \quad m \neq 0 \wedge n \neq 0. \quad (88)$$

Thus any decomposition

$$\hat{c} = {}^1\hat{k} + {}^2\hat{k} + {}^3\hat{k} + {}^4\hat{k} \quad (89)$$

of $\hat{c}$ into one-quadrant functions yields functions ${}^l\hat{k}$ that meet the conditions of Theorem 4 or Corollary 2. Consequently each of these decompositions corresponds to a set $\{{}^l k: l = 1, 2, 3, 4\}$ of recursively stable functions.

◆ Problem of an optimal solution

As Eqs. (71) and (72) show, the final solution $\{{}^l k: l = 1, 2, 3, 4\}$ may consist of functions with an infinite number of nonzero sample values. It would be advantageous to find such a decomposition of $\hat{c}$ which guarantees functions ${}^l k$ with a minimum finite number of nonzero sample values such that

$$1/C(z_1, z_2) = \prod_{l=1}^4 [1/K_l(z_1, z_2)] \quad (90)$$

is true, at least in a good approximation. This is also discussed in the following section on the truncation problem. The author does not have such a decomposition procedure to offer for $\hat{c}$. Some recommendations, however, seem reasonable. First, because $\hat{c}$ is centrally symmetric, the decomposition should be such that

$${}^1\hat{k}_{m,n} = {}^3\hat{k}_{-m,-n} \quad (91)$$

and

$${}^2\hat{k}_{m,n} = {}^4\hat{k}_{-m,-n}. \quad (92)$$

If, in addition, c (and consequently $\hat{c}$) is symmetric with respect to the axes $m = 0$ and $n = 0$, the following relations should hold:

$${}^1\hat{k}_{0,n} = {}^4\hat{k}_{0,n}, \quad n \geq 0, \quad (93)$$

$${}^2\hat{k}_{0,n} = {}^3\hat{k}_{0,n}, \quad n \leq 0, \quad (94)$$

$${}^1\hat{k}_{m,0} = {}^2\hat{k}_{m,0}, \quad m \geq 0, \quad (95)$$

$${}^3\hat{k}_{m,0} = {}^4\hat{k}_{m,0}, \quad m \leq 0. \quad (96)$$

It might be of interest that the problem of optimal de-

composition of $\hat{c}$ would become simple if

$$\hat{c}_{m,n} \equiv 0, \quad m \neq 0 \wedge n \neq 0. \quad (97)$$

Such cepstra occur if and only if c is separable, that is, if

$$c_{m,n} = s_m h_n. \quad (98)$$

Equation (98) means that the two-dimensional decomposition problem is reduced to the decomposition of a column vector s and a row vector h into minimum and maximum delay operators [6, 12]. These operators correspond to the cepstra

$$\{\hat{c}_{0,0}, \hat{c}_{1,0}, \hat{c}_{2,0}, \dots\}, \quad (99)$$

$$\{\dots, \hat{c}_{-2,0}, \hat{c}_{-1,0}, 0\}$$

and

$$\{0, \hat{c}_{0,1}, \hat{c}_{0,2}, \dots\}, \quad (100)$$

$$\{\dots, \hat{c}_{0,-2}, \hat{c}_{0,-1}, 0\}, \text{ respectively.}$$

◆ Truncation problem

As pointed out in the previous section, the decomposition of c may result in operators ${}^l k$ having an infinite number of sample points. If these operators are to be implemented numerically, some truncation of the operators becomes mandatory. This truncation means not only that the decomposition (90) becomes an approximate one but also that the recursive stability of the operators ${}^l k$ may be affected. Theorem 5 guarantees that there exist truncations for recursively stable operators ${}^l k$ such that the resulting operators ${}^l k$ remain recursively stable.

Theorem 5

We assume that the power series

$$B_1(z_1, z_2) = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} {}^1b_{m,n} z_1^m z_2^n \quad (101)$$

is absolutely convergent and nonzero for all $(z_1, z_2) \in D_1$. There then exists a pair M_1, N_1 such that

$$|\dot{B}_1(z_1, z_2)| = \sum_{m=0}^{m_1} \sum_{n=0}^{n_1} {}^1b_{m,n} z_1^m z_2^n > 0 \quad (102)$$

for all $(m_1, n_1) \in \{(m_1, n_1): m_1 > M_1 \wedge n_1 > N_1\}$ and for all $(z_1, z_2) \in D_1$.

The proof of this theorem is based on the uniform convergence of the power series (101). The extension of the theorem to functions other than those in the first quadrant is straightforward.

To find constraints on the truncation parameters to be applied on ${}^l k$, in which $l = 1, 2, 3, 4$, we consider $(M_1 + 1) \times (N_1 + 1)$ subarrays of 1k and 3k ; $(M_2 + 1) \times (N_2 + 1)$ subarrays of 2k and 4k ; and a $(4M_c + 1) \times (4N_c + 1)$ array c . The cascaded convolution of the subarrays yields an array with $[2(M_1 + M_2) + 1] \times [2(N_1 + N_2) + 1]$ sample points. It contains an array as large as c , if

$$(M_1 + M_2) \geq 2M_c \quad (103)$$

and

$$N_1 + N_2 \geq 2N_c. \quad (104)$$

For arrays c symmetric with respect to the axes $m = 0$ and $n = 0$, the relations (93) – (96) apply. In this case the constraints on the truncation parameters are

$$M_1 = M_2 \geq M_c, \quad (105)$$

$$N_1 = N_2 \geq N_c. \quad (106)$$

It is our experience [11] that the truncation conditions

$$M_1 = M_c, \quad N_1 = N_c \quad (107)$$

may give satisfactory results. If Eqs. (107) hold, the decomposition of c means a saving of filter coefficients which ranges from 36 percent ($M_c = N_c = 1$) to 75 percent ($M_c \rightarrow \infty \wedge N_c \rightarrow \infty$).

• Numerical implementation

This section comments on the computation of $\hat{c}$, on the decomposition procedure, and on the application of the decomposed recursive filter.

To determine $\hat{c}$, Eqs. (36), (81), and (83) must be evaluated. This is done by Fast Fourier Transform (FFT) techniques [16] as indicated in Fig. 2. The functions $\{C_{m,n}\}$ and $\{\hat{C}_{m,n}\}$ are sampled versions of $C'(u, v)$ and $\hat{C}'(u, v)$, whereas $\hat{\zeta}$ is an aliased version of the desired result $\hat{c}$ [12]. The degree of aliasing can be controlled by the rate at which $C'(u, v)$ is sampled.

The function $\hat{\zeta}$, truncated at the Nyquist subscripts $\pm m_{Ny}$ and $\pm n_{Ny}$, is decomposed according to

$$\hat{\zeta}_{tr} = {}^1\hat{\kappa} + {}^2\hat{\kappa} + {}^3\hat{\kappa} + {}^4\hat{\kappa}, \quad (108)$$

where $\hat{\kappa}$ are approximations of operators k .

Since $\hat{\zeta}_{tr}$ exhibits the same symmetries as $\hat{c}$, the decomposition should be performed according to (91) and (92). This decomposition has the advantage that only ${}^1\hat{\kappa}$ and ${}^4\hat{\kappa}$ have to be determined. The values ${}^3\hat{\kappa}$ and ${}^2\hat{\kappa}$ are obtained by rotating the subscript planes through 180° .

The computation of an $(M_1 + 1) \times (N_1 + 1)$ array ${}^1\hat{\kappa}$ from ${}^1\hat{\kappa}$ should be performed in three steps. First ${}^1\hat{\kappa}_{0,0}$ is obtained by Eq. (73). Then the row vector

$$\{{}^1\hat{\kappa}_{0,n}\}_{1 \leq n \leq N_1}$$

is determined by Eq. (72), and finally the submatrix

$$\{{}^1\hat{\kappa}_{m,n}\}_{\substack{1 \leq m \leq M_1 \\ 0 \leq n \leq N_1}}$$

is obtained by Eq. (71). The same procedure is applied to

$$\{{}^4\hat{\kappa}_{-m,n}\}_{\substack{m \leq 0 \\ n \leq 0}}$$

to obtain

$$\{{}^4\hat{\kappa}_{-m,n}\}_{\substack{m \leq 0 \\ n \geq 0}}$$

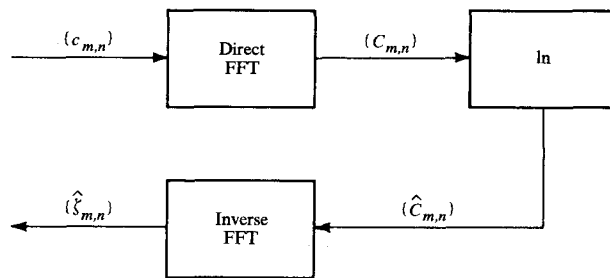


Figure 2 Block diagram of the approximate cepstrum transform for evaluation of Eqs. (36), (81), and (83). The transformation in the first step of the decomposition procedure.

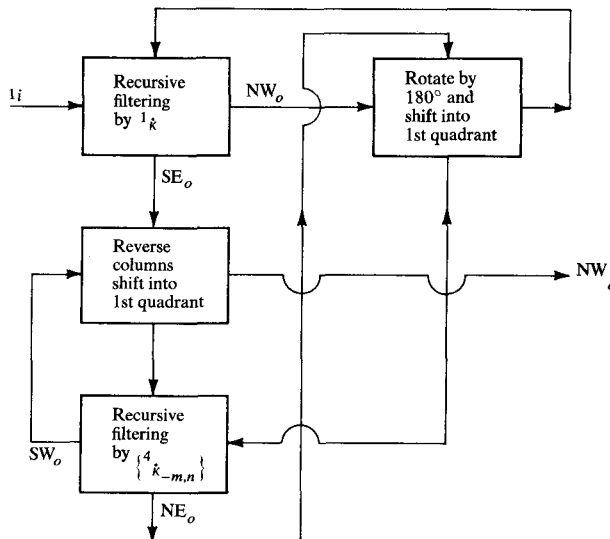


Figure 3 Block diagram of the decomposition procedure.

The block diagram in Fig. 3 resumes the decomposition of c .

It should be recalled that the result only approximates the sought functions ${}^l k$. The approximation errors stem from the aliasing exhibited in $\hat{\zeta}$ and also from the truncation applied to ${}^l \kappa$, $l = 1, 2, 3, 4$.

According to the discussions of Eqs. (22), (23), and (24), the input-output relation

$$\begin{aligned} O(z_1, z_2) &= I_1(z_1, z_2) / C(z_1, z_2) \\ &\approx ([I_1(z_1, z_2) / \dot{\kappa}_1(z_1, z_2)] / \dot{\kappa}_2 \\ &\quad \times (z_1, z_2)) / \dot{\kappa}_3(z_1, z_2) / \dot{\kappa}_4(z_1, z_2) \end{aligned} \quad (109)$$

can be evaluated only by implementing the causal recursive algorithm. Figure 4 demonstrates how this can be done. It is clear (see [2]) that the computation of the intermediate results is to be stopped after the output values have decayed to reasonably small numbers.

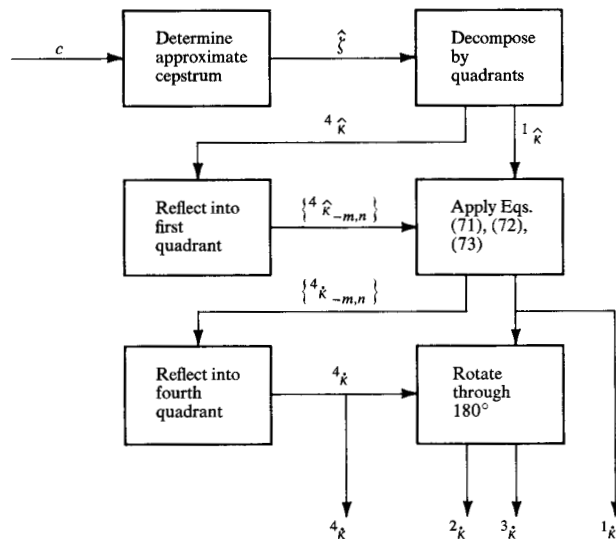


Figure 4 Implementation of the decomposed recursive filter $1/C(z_1, z_2)$. The superscripts of the intermediate filter results indicate the proper position of the upper-left corner of the array.

Nonrecursive solutions

Instead of decomposing $1/C(z_1, z_2)$ into a cascade of recursive filters [see Eq. (90)], a decomposition into convolutional filters might be preferred:

$$1/C(z_1, z_2) = F_1(z_1, z_2)F_2(z_1, z_2)F_3(z_1, z_2)F_4(z_1, z_2). \quad (110)$$

The corresponding functions $^l f$ can also be determined from the cepstra $^l \hat{k}$, $l = 1, 2, 3, 4$. It is sufficient to demonstrate this for $^1 k$ and $^1 f$. The operator $^1 k$ is assumed to be recursively stable. From

$$\ln(1/K_1(z_1, z_2)) = \ln F_1(z_1, z_2) \quad (111)$$

we have

$$-^1 \hat{k} = ^1 \hat{f}. \quad (112)$$

Thus, replacing $^1 \hat{b}$ by $(-^1 \hat{k})$ and $^1 b$ by $^1 f$ in Eqs. (71), (72), and (73), we find

$$^1 f_{0,0} = \exp(-^1 \hat{k}_{0,0}), \quad (113)$$

$$^1 f_{p,q} = -\sum_{m=1}^p \sum_{n=0}^q (m/p) \hat{k}_{m,n} f_{p-m, q-n}, \quad p \neq 0, \quad (114)$$

$$^1 f_{p,q} = -\sum_{m=0}^p \sum_{n=1}^q (n/q) \hat{k}_{m,n} f_{p-m, q-n}, \quad q \neq 0. \quad (115)$$

The decomposition (110) is preferred over (90) if it requires a lower number of filter weights for the same accuracy.

Alternatively to Eqs. (113), (114), and (115), the least-squares approximations of $^l f$ could be determined from $^l k$, $l = 1, 2, 3, 4$, by the Wiggins algorithm [2].

Numerical examples

As pointed out in the previous section, the problem of the optimal decomposition of a recursive operator c consists in finding an appropriate decomposition of $\hat{c}$ and an appropriate truncation of the resulting operators. In this section it is demonstrated that the procedure characterized by the decomposition rules (93) and (96) and the truncation rules (105)–(107) may be the optimal decomposition procedure. We also discuss examples of unsatisfactory results obtained by these rules.

The operators c that we discuss are synthesized from one-quadrant operators with 2×2 samples. The reasons are that

1. The optimal decomposition is, a priori, known exactly in two cases.
2. Since the occurring one-quadrant operators have only a few samples (2×2 , 3×3 , 4×4), their recursive stability can readily be checked.

An example of practical interest could also have been discussed, namely the computation of recursive filters for the enhancement of radioscinographic images [11, 17]. The decomposition rules mentioned here were able to specify a near-optimal solution for this problem. However, this example is not well suited to point out the potential and shortcomings of these rules.

Example 1

Consider a function c ,

$$c = ^1 k * ^2 k * ^3 k * ^4 k, \quad (116)$$

where

$$^1 k_{m,n} = ^2 k_{m,-n} = ^3 k_{-m,-n} = ^4 k_{-m,n}, \quad m \geq 0, n \geq 0. \quad (117)$$

With the recursively stable function [4]

$$^1 k = \begin{bmatrix} 1.00 & 0.90 \\ 0.90 & 0.85 \end{bmatrix} \leftarrow, \quad (118)$$

↑

we have

$$c = \begin{bmatrix} 0.68850 & 2.7639 & 4.15082 & 2.7639 & 0.68850 \\ 2.76390 & 11.0930 & 16.65830 & 11.0930 & 2.76390 \\ 4.15082 & 16.6583 & 25.01520 & 16.6583 & 4.15082 \\ 2.76390 & 11.0930 & 16.65830 & 11.0930 & 2.76390 \\ 0.68850 & 2.7639 & 4.15082 & 2.7639 & 0.68850 \end{bmatrix} \leftarrow. \quad (119)$$

↑

In both (118) and (119) the vertical and horizontal arrows indicate the axes $m = 0$ and $n = 0$, respectively.

From the construction of c it is clear that $\hat{c}$ is to be decomposed according to Eqs. (93)–(96). In addition, the 3×3 result

$${}^1\dot{\kappa} = \begin{bmatrix} 1.0000 & 0.9000 & -0.9\text{E-}6 \\ 0.9000 & 0.8500 & -0.6\text{E-}6 \\ -0.4\text{E-}6 & 0.2\text{E-}6 & -0.8\text{E-}7 \end{bmatrix} \leftarrow \begin{matrix} \uparrow \end{matrix} \quad (120)$$

shows that the truncation rules (105)–(107) are applicable, since ${}^1\dot{\kappa}$ is identical to ${}^1\kappa$ except for entries in the third row and column, which may be neglected.

• Example 2

We modify the operator (119) by adding the spike function 0.011δ . That means $c_{0,0} = 25.0152$ is replaced by 25.2903. Contrary to example 1, no decomposition of c is now known a priori. However, because this c exhibits the same symmetries as the function c in the first example, it can be expected that Eqs. (93)–(96) describe the optimal decomposition of $\hat{c}$. Similar to Example 1 the decomposition can be characterized by one operator, e.g., ${}^1\dot{\kappa}$. As a 4×4 result we find

$${}^1\dot{\kappa} = \begin{bmatrix} 1.4037 & 0.7090 & -0.0033 & -0.0069 \\ 0.7090 & 0.7916 & 0.0305 & -0.0183 \\ -0.0033 & 0.0305 & 0.0210 & -0.0075 \\ -0.0069 & -0.0183 & -0.0075 & 0.0024 \end{bmatrix} \leftarrow \begin{matrix} \uparrow \end{matrix} \quad (121)$$

The operator ${}^1\dot{\kappa}$ is recursively stable since it can be shown that the z transform of (121) is nonzero for all $(z_1, z_2) \in D_1$. The proof is based on the properties of the upper left 2×2 subarray of ${}^1\dot{\kappa}$.

Equation (121) does not have such a distinct cutoff beyond $m = 1, n = 1$ as does (120). Thus the truncation rules (105)–(107) cannot be expected to yield operators ${}^l\dot{\kappa}$, $l = 1, 2, 3, 4$, which, when convolved in a cascade, would result in a good approximation c_{appr} of c . Even with 4×4 operators, the approximation is not quite satisfactory, as a comparison of c and a subarray of $e = c_{\text{appr}} - c$ demonstrates [See Eq. (122)].

Because the operators ${}^l\dot{\kappa}$ are to be used for recursive filtering, the approximation error function e might be inappropriate to characterize the quality of the decomposition of c . Alternatively, c can be recursively filtered by these operators. If the decomposition is a good one, the result $\hat{o}$ should resemble a delta function. As the submatrix (123) demonstrates, the solution given by Eq. (121) does this job fairly well.

• Example 3

In this example a 5×5 function c is presented which can be decomposed into four 2×2 operators. However, the

$e_{\text{sub}} =$

$$\begin{bmatrix} 0.159264\text{E-}02 & 0.105697\text{E-}02 & 0.239717\text{E-}03 & -0.403976\text{E-}04 & -0.189802\text{E-}04 & 0.851596\text{E-}07 & 0.160910\text{E-}06 \\ -0.264569\text{E-}02 & -0.143818\text{E-}02 & 0.357122\text{E-}04 & 0.171492\text{E-}03 & 0.145821\text{E-}04 & -0.189372\text{E-}05 & 0.851477\text{E-}07 \\ -0.147880\text{E-}00 & -0.104905\text{E-}00 & -0.312095\text{E-}01 & 0.193911\text{E-}02 & 0.219917\text{E-}02 & 0.145812\text{E-}04 & -0.189804\text{E-}04 \\ -0.200212\text{E-}00 & -0.145703\text{E-}00 & -0.309612\text{E-}01 & -0.423922\text{E-}02 & -0.193903\text{E-}02 & 0.171491\text{E-}03 & -0.403971\text{E-}04 \\ -0.749245\text{E-}01 & -0.556574\text{E-}01 & -0.397092\text{E-}01 & -0.509626\text{E-}01 & -0.312099\text{E-}01 & 0.357172\text{E-}04 & 0.239724\text{E-}03 \\ 0.172424\text{E-}02 & -0.724792\text{E-}04 & -0.556641\text{E-}01 & -0.145707\text{E-}00 & -0.104906\text{E-}00 & -0.143816\text{E-}02 & 0.105699\text{E-}02 \\ 0.474548\text{E-}02 & -0.170898\text{E-}02 & -0.749350\text{E-}01 & -0.200217\text{E-}00 & -0.147882\text{E-}00 & -0.264567\text{E-}02 & 0.159267\text{E-}02 \\ -0.175476\text{E-}02 & -0.734329\text{E-}04 & -0.556622\text{E-}01 & -0.145707\text{E-}00 & -0.104906\text{E-}00 & -0.143817\text{E-}02 & 0.105699\text{E-}02 \\ -0.749216\text{E-}01 & -0.556555\text{E-}01 & -0.397084\text{E-}01 & -0.509624\text{E-}01 & -0.312098\text{E-}01 & 0.357162\text{E-}04 & 0.239723\text{E-}03 \\ -0.200212\text{E-}00 & -0.145703\text{E-}00 & -0.309611\text{E-}01 & -0.423913\text{E-}02 & 0.193906\text{E-}02 & 0.171490\text{E-}03 & -0.403977\text{E-}04 \\ -0.147880\text{E-}00 & -0.104905\text{E-}00 & -0.312094\text{E-}01 & 0.193914\text{E-}02 & 0.219918\text{E-}02 & 0.145810\text{E-}04 & -0.189804\text{E-}04 \\ -0.264569\text{E-}02 & -0.143818\text{E-}02 & 0.357112\text{E-}04 & 0.171492\text{E-}03 & 0.145820\text{E-}04 & -0.189371\text{E-}05 & 0.851517\text{E-}07 \\ 0.159264\text{E-}02 & 0.105697\text{E-}02 & 0.239716\text{E-}03 & -0.403980\text{E-}04 & -0.189803\text{E-}04 & 0.851621\text{E-}07 & 0.160911\text{E-}06 \end{bmatrix} \leftarrow \begin{matrix} \uparrow \end{matrix}$$

(122)

$\hat{o}_{\text{sub}} =$

$$\begin{bmatrix} -0.613363\text{E-}02 & -0.474950\text{E-}02 & -0.159342\text{E-}02 & 0.143817\text{E-}03 & 0.139719\text{E-}03 & -0.713078\text{E-}04 & 0.121408\text{E-}04 \\ 0.864866\text{E-}02 & 0.617515\text{E-}02 & 0.160198\text{E-}02 & -0.367183\text{E-}03 & -0.787302\text{E-}04 & 0.994637\text{E-}04 & -0.522314\text{E-}04 \\ -0.113625\text{E-}01 & -0.759114\text{E-}02 & -0.138257\text{E-}02 & -0.597621\text{E-}03 & -0.646202\text{E-}04 & -0.825182\text{E-}04 & 0.954837\text{E-}04 \\ 0.137227\text{E-}01 & 0.861436\text{E-}02 & 0.873944\text{E-}03 & -0.743999\text{E-}03 & 0.291746\text{E-}03 & -0.321593\text{E-}04 & -0.957433\text{E-}04 \\ -0.143766\text{E-}01 & -0.852288\text{E-}02 & -0.121410\text{E-}03 & 0.554047\text{E-}03 & -0.524485\text{E-}03 & 0.280924\text{E-}03 & -0.319643\text{E-}04 \\ -0.111293\text{E-}01 & 0.641693\text{E-}02 & -0.559615\text{E-}03 & -0.187813\text{E-}03 & 0.497826\text{E-}03 & -0.529216\text{E-}03 & 0.295783\text{E-}03 \\ -0.826582\text{E-}03 & -0.602300\text{E-}03 & -0.240482\text{E-}03 & 0.724608\text{E-}06 & -0.176470\text{E-}03 & 0.648627\text{E-}03 & -0.738702\text{E-}03 \\ 0.130282\text{E-}02 & 0.475957\text{E-}03 & 0.530685\text{E-}03 & -0.218473\text{E-}03 & -0.609808\text{E-}03 & -0.676552\text{E-}04 & 0.817899\text{E-}03 \\ -0.292059\text{E-}03 & -0.172962\text{E-}02 & 0.405336\text{E-}03 & -0.462968\text{E-}03 & 0.624410\text{E-}02 & -0.829096\text{E-}02 & 0.832244\text{E-}02 \\ -0.100263\text{E-}01 & -0.246092\text{E-}03 & 0.113392\text{E-}02 & -0.619821\text{E-}03 & 0.167933\text{E-}01 & -0.139117\text{E-}01 & 0.131143\text{E-}01 \\ -0.178754\text{E-}03 & -0.158190\text{E-}02 & 0.431565\text{E-}03 & -0.276147\text{E-}03 & 0.601419\text{E-}02 & -0.792788\text{E-}02 & 0.776092\text{E-}02 \\ 0.110793\text{E-}02 & 0.248553\text{E-}03 & 0.222650\text{E-}03 & -0.137941\text{E-}03 & -0.891139\text{E-}03 & 0.356984\text{E-}03 & 0.237607\text{E-}03 \\ -0.458554\text{E-}03 & -0.128206\text{E-}03 & 0.348056\text{E-}04 & 0.243648\text{E-}03 & -0.321219\text{E-}03 & 0.814399\text{E-}03 & -0.966814\text{E-}03 \\ -0.105644\text{E-}01 & 0.573408\text{E-}02 & -0.107241\text{E-}02 & -0.347991\text{E-}03 & 0.494597\text{E-}03 & -0.497626\text{E-}03 & 0.315906\text{E-}03 \\ -0.135230\text{E-}01 & -0.750595\text{E-}02 & 0.575967\text{E-}03 & 0.846770\text{E-}03 & -0.538528\text{E-}03 & 0.219587\text{E-}03 & 0.955981\text{E-}05 \\ -0.124684\text{E-}01 & 0.715635\text{E-}02 & -0.334558\text{E-}04 & -0.965414\text{E-}03 & 0.365733\text{E-}03 & -0.199897\text{E-}05 & -0.117244\text{E-}03 \\ -0.956713\text{E-}02 & -0.558962\text{E-}02 & -0.232851\text{E-}03 & 0.800152\text{E-}03 & -0.181203\text{E-}03 & -0.965190\text{E-}04 & 0.113540\text{E-}03 \\ -0.615787\text{E-}02 & 0.355146\text{E-}02 & 0.229146\text{E-}03 & -0.515932\text{E-}03 & 0.734335\text{E-}04 & 0.254887\text{E-}04 & -0.740222\text{E-}04 \\ -0.286662\text{E-}02 & -0.155308\text{E-}02 & -0.903530\text{E-}04 & 0.221738\text{E-}03 & -0.299074\text{E-}04 & -0.449883\text{E-}04 & 0.321043\text{E-}04 \end{bmatrix} \leftarrow \begin{matrix} \uparrow \end{matrix}$$

(123)

decomposition procedure (93)–(96) is inadequate to provide these operators. Let

$$c = \begin{bmatrix} 0.72250 & 2.8305 & 4.14922 & 2.6973 & 0.65610 \\ 2.83050 & 11.2267 & 16.65830 & 10.9593 & 2.69730 \\ 4.14922 & 16.6583 & 25.01840 & 16.6583 & 4.14922 \\ 2.69730 & 10.9593 & 16.65830 & 11.2267 & 2.83050 \\ 0.65610 & 2.6973 & 4.14922 & 2.8305 & 0.72250 \end{bmatrix} \leftarrow \begin{matrix} \uparrow \\ (124) \end{matrix}$$

The function c has been constructed by the cascaded convolution

$$c = {}^1k * {}^1k * {}^3k * {}^3k, \quad (125)$$

where

$${}^1k_{m,n} = {}^3k_{-m,-n}, \quad m \geq 0, n \geq 0, \quad (126)$$

and 1k is given by Eq. (118).

From Eq. (125) it is evident that our procedure is not adequate to find the optimal decomposition of c . In the cepstrum domain this is reflected by the fact that $\hat{c}$ is identically zero for all

$$(m, n) \in \{(m, n): m > 0 \wedge n < 0 \vee m < 0 \wedge n > 0\}. \quad (127)$$

For illustration a subarray of the approximate cepstrum $\hat{\xi}$ is given here.

$$\hat{\xi}_{\text{sub}} = \begin{bmatrix} -0.3280506 & 0.0000012 & -0.0000008 & -0.0000002 & 0.0000004 \\ 0.4860004 & -0.0000013 & 0.0000008 & 0.0000001 & -0.0000003 \\ -0.8100015 & 0.0000028 & -0.0000022 & 0.0000013 & -0.0000011 \\ 1.7999992 & -0.0000015 & 0.0000008 & 0.0000001 & -0.0000003 \\ -0.0000005 & 1.7999954 & -0.8099970 & 0.4859967 & -0.3280469 \\ 1.7999992 & 0.0799982 & -0.0719988 & 0.0647995 & -0.0583197 \\ -0.8100015 & -0.0179969 & 0.0631976 & -0.0554384 & 0.0485987 \\ 0.4860004 & 0.0647983 & -0.0554389 & 0.0473463 & -0.0403558 \\ -0.3280506 & -0.0583183 & 0.0485989 & -0.0403555 & 0.0333769 \end{bmatrix} \leftarrow \begin{matrix} \uparrow \\ (128) \end{matrix}$$

In spite of the aforementioned inadequacy it is interesting to decompose c using Eqs. (93)–(96). This means that we look for a decomposition according to

$$c = {}^1k' * {}^2k' * {}^3k' * {}^4k' \quad (129)$$

rather than according to Eq. (125). It can be concluded from (125) and (127) that ${}^2k'$ and ${}^4k'$ are 2×2 operators, where

$${}^2k_{m,n}' = {}^1k_{m,0} {}^3k_{0,n} \quad (130)$$

and

$${}^4k_{m,n}' = {}^2k_{m,-n}'. \quad (131)$$

The operators ${}^1k'$ and ${}^3k'$, however, may be of infinite size since ${}^1k'$ could be obtained by recursively filtering ${}^1k * {}^1k$ with

$$\{{}^2k_{m,-n}'\}_{m \geq 0, n \leq 0} \quad (132)$$

The numerical results are in accordance with these facts. Operator 2k has a distinct cutoff beyond $m = 1, n = -1$, whereas 1k slowly decays in the lower right submatrix, thus making a larger matrix desirable:

$${}^1k = \begin{bmatrix} 1.000000 & 0.899998 & -0.536E-6 & -0.238E-6 \\ 0.899999 & 0.889996 & -0.113E-5 & 0.477E-6 \\ -0.113E-5 & 0.358E-6 & 0.001600 & -0.001440 \\ -0.477E-6 & 0.596E-6 & -0.001440 & 0.001296 \end{bmatrix} \leftarrow \begin{matrix} \uparrow \\ (133) \end{matrix}$$

$${}^2k = \begin{bmatrix} -0.238E-6 & -0.536E-6 & 0.899998 & 1.000000 \\ 0.617E-6 & -0.100E-5 & 0.809996 & 0.899999 \\ 0.772E-7 & -0.185E-6 & 0.458E-6 & -0.113E-5 \\ 0.249E-6 & -0.442E-7 & 0.812E-6 & -0.477E-6 \end{bmatrix} \leftarrow \begin{matrix} \uparrow \\ (134) \end{matrix}$$

Nevertheless, the performance of this solution is not so far from the optimum. This property can be seen from the arrays e_{sub} [Eq. (135)] and ϕ_{sub} [Eq. (136)], which have the same meaning as in Example 2.

$e_{\text{sub}} =$

$$\begin{bmatrix} -0.726528E-09 & -0.300764E-09 & -0.322484E-09 & -0.620874E-13 & -0.106655E-12 & 0.486984E-19 & 0.282767E-19 \\ -0.814738E-09 & -0.734651E-10 & -0.352007E-09 & -0.437013E-12 & -0.488424E-12 & 0.381983E-18 & 0.448300E-19 \\ -0.104863E-02 & 0.102936E-05 & 0.804396E-06 & 0.568578E-06 & 0.201465E-06 & -0.211685E-12 & -0.531997E-13 \\ -0.198347E-02 & 0.459036E-06 & 0.162494E-05 & 0.170333E-06 & 0.666931E-06 & -0.520017E-12 & -0.167008E-13 \\ -0.987053E-03 & 0.324249E-04 & -0.645691E-05 & 0.149511E-05 & 0.111582E-05 & -0.194287E-09 & -0.322311E-09 \\ -0.183105E-03 & -0.120163E-03 & -0.400543E-04 & -0.105484E-05 & 0.149877E-05 & -0.218226E-10 & -0.458015E-09 \\ -0.244141E-03 & -0.183105E-03 & -0.998497E-03 & -0.198635E-02 & -0.104823E-02 & 0.102559E-08 & -0.649578E-09 \\ -0.167847E-03 & -0.123978E-03 & -0.202656E-02 & -0.430578E-02 & -0.233192E-02 & 0.178572E-08 & -0.208588E-09 \\ -0.988007E-03 & -0.202274E-02 & -0.135297E-02 & -0.727915E-03 & -0.445602E-03 & 0.304066E-09 & 0.108114E-08 \\ -0.198347E-02 & -0.430326E-02 & -0.726894E-03 & 0.348490E-02 & 0.188940E-02 & -0.143340E-08 & 0.469635E-09 \\ -0.104864E-02 & -0.233199E-02 & -0.445418E-03 & 0.188950E-02 & 0.104969E-02 & -0.817241E-09 & -0.278076E-09 \\ -0.814738E-09 & 0.542428E-09 & 0.229420E-08 & -0.391476E-09 & -0.170395E-08 & 0.192376E-14 & -0.626138E-16 \\ -0.726528E-09 & -0.307082E-09 & 0.148689E-08 & 0.446240E-09 & -0.556149E-09 & 0.495778E-16 & 0.147329E-15 \end{bmatrix} \leftarrow \begin{matrix} \uparrow \\ (135) \end{matrix}$$

0.280967E-02	0.504769E-05	0.624132E-05	-0.154875E-03	0.714838E-03	-0.105378E-02	0.119510E-02
-0.312973E-02	0.308633E-04	-0.490645E-04	0.217502E-03	-0.835808E-03	0.120684E-02	-0.135821E-02
-0.341518E-02	-0.385113E-04	0.772769E-04	-0.276761E-03	0.961070E-03	-0.136687E-02	0.152817E-02
-0.354431E-02	-0.892836E-04	0.404467E-05	0.247829E-03	-0.101446E-02	0.146519E-02	-0.164298E-02
-0.332886E-02	0.519752E-03	-0.343769E-03	0.272589E-05	0.875956E-03	-0.139354E-02	0.160498E-02
-0.250447E-02	-0.148699E-02	0.115089E-02	-0.662563E-03	-0.376550E-03	0.999450E-03	-0.127663E-02
0.697545E-03	0.332353E-02	-0.272302E-02	0.199929E-02	-0.724934E-03	-0.653226E-04	0.461661E-03
0.164124E-02	-0.561980E-02	0.469004E-02	-0.368012E-02	0.212960E-02	-0.114058E-02	0.599022E-03
-0.525562E-02	0.904012E-02	-0.764826E-02	0.624158E-02	-0.432133E-02	0.305477E-02	-0.229894E-02
0.100776E-01	-0.114254E-01	0.965816E-02	-0.794142E-02	0.573437E-02	-0.426973E-02	0.337229E-02
-0.109169E-01	0.143818E-01	-0.121632E-01	0.100813E-01	-0.754331E-02	0.584474E-02	-0.477361E-02
0.861964E-02	-0.114250E-01	0.837923E-02	-0.561539E-02	0.377813E-02	-0.267009E-02	0.207850E-02
-0.649787E-02	0.874132E-02	-0.303256E-02	0.179719E-02	-0.643940E-03	0.827004E-04	0.863849E-04
0.438516E-02	-0.461432E-02	0.321013E-02	-0.719524E-03	-0.411731E-05	0.267322E-03	-0.248437E-03
-0.298584E-02	0.437045E-02	-0.212724E-02	0.266846E-03	0.143739E-03	-0.210926E-03	0.884780E-04
0.208619E-02	-0.316373E-02	0.151765E-02	-0.178681E-03	-0.225013E-04	-0.193318E-04	0.178318E-03
-0.149566E-02	0.229929E-02	-0.116076E-02	0.247772E-03	-0.170123E-03	0.251862E-03	-0.398289E-03
0.102591E-02	-0.156502E-02	0.855297E-03	-0.290920E-03	0.272346E-03	-0.347115E-03	0.452698E-03
-0.544717E-03	0.816697E-03	-0.477380E-03	0.208324E-03	-0.208049E-03	0.249972E-03	-0.303273E-03

(136)

Concluding remarks

An unstable recursive filter with a finite number of coefficients always has an infinite number of decompositions into four stable recursive filters, provided it has a nonzero frequency response with zero phase. The decomposability is proved in this paper by means of a new criterion that relates the stability of a recursive filter to the properties of its cepstrum. One might say that this stability criterion, compared to that of Shanks, has the advantage of being constructive, i.e., it permits the specified filter to be decomposed as exactly as is desired.

Two questions remain unanswered:

1. Under what suppositions is it possible to decompose exactly an unstable recursive filter into stable recursive filters that have a minimum finite number of coefficients; and what is the number of these coefficients?
2. If such a decomposition exists, how can it be found?

In spite of these unanswered questions, the results given here are of practical value, as can be seen from the application of two-dimensional recursive filters to radio-scintigraphic images [11, 17]. These filters have their theoretical foundation in the results of this paper.

Acknowledgments

The author thanks W. Kattwinkel, manager of the IBM Deutschland Heidelberg Scientific Center when this work was being done, for his support and encouragement. He also gratefully acknowledges helpful discussions with H. G. Meder and G. Jaeschke. Especially worthy of mention is the gratifying cooperation of Mrs. M. Gamp in preparing the manuscript.

References and notes

1. J. L. Shanks, "Two-dimensional Recursive Filters," in *SWIECO Record of Technical Papers* 21, 19E1 (1969).
2. J. L. Shanks and J. H. Justice, "Stability and Synthesis of Two-dimensional Recursive Filters," *IEEE Trans. Audio and Electroacoustics* AU-20, 115 (1972).
3. C. Farmer and J. B. Bednar, "Stability of Spatial Digital Filters," *Math. Biosci.* 14, 113 (1972).
4. T. S. Huang, "Stability of Two-dimensional Recursive Filters," *IEEE Trans. Audio and Electroacoustics* AU-20, 158 (1972).
5. The original form of the cepstrum transformation was the power spectrum of the logarithm of a power spectrum. See B. P. Bogert, M. J. B. Healy, and J. W. Tukey, "The Quefrency Analysis of Time Series for Echoes," in *Proc. Symp. on Time Series Analysis*, M. Rosenblatt, ed., John Wiley and Sons, Inc., New York, 1963, Ch. 15, p. 209. Throughout the present paper the (complex) cepstrum transformation is used as introduced by Oppenheim et al. [12].
6. J. L. Shanks, "Recursion Filters for Digital Processing," *Geophysics* 32, 33 (1967).
7. E. A. Robinson, *Statistical Communication and Detection*, Hafner Press, New York, 1967, Chs. 6 and 7.
8. *Ibid.*, App. 2.
9. The positive direction of the m axis is downward, and the n axis points to the right. Compare with Fig. 1. This choice has been made following the usual subscripting of matrix elements (see also [1, 2]). In another convention [4] the m axis points to the right and the n axis is directed downward.
10. Usually, recursive filters are defined as $A(z_1, z_2)/B(z_1, z_2)$. However, because $1/(z_1, z_2) \cdot A(z_1, z_2)$ can be interpreted as the input to a recursive filter $F(z_1, z_2) = 1/B(z_1, z_2)$, $A(z_1, z_2) = 1$ may be assumed without loss of generality.
11. P. Pistor, "Digital Processing of Scintigraphic Images by Two-dimensional Recursive Wiener Filters," Report 70.12.006, IBM Deutschland Heidelberg Scientific Center, 1970.
12. A. V. Oppenheim, R. W. Schaffer, and T. G. Stockham, Jr., "Nonlinear Filtering of Multiplied and Convolved Signals," *Proc. IEEE* 56, 1264 (1968).
13. W. I. Smirnow, "Lehrgang der höheren Mathematik, Teil III, 2," *VEB Deutscher Verlag der Wissenschaften*, Berlin, 1967, Ch. 4.
14. *Ibid.*, Ch. 1.
15. G. P. Tolstov, *Fourier Series*, Prentice Hall, Inc., Englewood Cliffs, N.J., 1962, p. 178.
16. W. T. Cochran et al., "What is the Fast Fourier Transform?" *Proc. IEEE* 55, 1664 (1967).
17. P. Pistor et al., "Digital Image Processing in Nuclear Medicine," *Kerntechnik, Isotopentechnik und -Chemie* 14, 299–306 and 353–359 (1972).

Received February 5, 1973

The author is located at the IBM Deutschland Scientific Center, Tiergartenstrasse 15, 6900 Heidelberg, Germany.