Circuits Syst Signal Process (2018) 37:1669–1691
https://doi.org/10.1007/s00034-017-0618-2

Synthesis and Realization of 3-D Orthogonal FIR Filters Using Pipeline Structures

P. Poczekajło[1] · R. T. Wirski[1]

Received: 5 December 2016 / Revised: 19 July 2017 / Accepted: 19 July 2017 /
Published online: 4 August 2017

Abstract The authors present a novel design algorithm for 3-D orthogonal filters. Both separable and non-separable cases are discussed. In the separable case, the synthesis leads to a cascade connection of 1-D systems. In the latter case, one obtains 2-D systems followed by a 1-D one. Realization techniques for these systems are presented which utilize Givens rotations and delay elements. The results are illustrated by examples of separable and non-separable 3-D system designs, i.e., Gaussian and Laplacian filters.

Keywords 3-D · Orthogonal FIR filter · Synthesis · Pipeline structure · Givens rotation

1 Introduction

Since the first pulse-code modulation transmission of digitally quantized speech, in World War II, digital signal processing (DSP) began to proliferate to all areas of human life. A classic DSP is based on linear systems described by impulse response functions and transfer functions implemented by structures built from adders, multipliers, and unit delays. Another approach was initiated by [31], known as the state space approach. It was also extended to the 2-D case by Roesser [18], as well as to three-dimensional (3-D) [9]. The steady increase in computational power encourages applying DSP techniques to multidimensional processing. However, the n-dimensional (n-D) DSP development has encountered difficulties caused by n-D polynomials [5]. Namely,

✉ P. Poczekajło
pawel.poczekajlo@tu.koszalin.pl

R. T. Wirski
robert.wirski@tu.koszalin.pl

[1] Faculty of Electronics and Computer Science, Koszalin University of Technology, Koszalin, Poland

there is no straightforward generalization of the fundamental theorem of algebra to higher dimensions. Classical digital systems are known to possess poor parameters under finite-precision arithmetic, like the sensitivity of the frequency response to changes in the structural parameters, noise, intrinsic oscillations, and limit cycles. These effects have led to the invention of wave filters [6] and orthogonal filters [2,3]. The most common approach to orthogonal filter synthesis is a transfer function decomposition and the state space approach. When it comes to multidimensional DSP, the former technique is of a limited use due to the n-D polynomials. In contrast, the latter provides an opportunity to extend 1-D state space techniques to higher dimensions, thanks to the 2-D, 3-D, and possibly n-D state space equations. The state space approach to lossless systems was initiated by the famous paper [30], where paraunitary matrix synthesis techniques were developed for the 2-D transfer function of a continuous system. The state space approach was also used to develop 2-D orthogonal filter synthesis [16] and simplified to cover a class of separable-denominator orthogonal filters [28] which found to be useful in real-time processing [21].

Nowadays, one can observe that a processed data becomes n-D like video, multichannel audio, machine vision, to name a few. The 3-D processing is especially important in medicine [1] and image/video processing [5,10] but also finds applications in other areas like material structures [12].

DSP synthesis is based on difference equations usually transformed by the $\mathcal{Z}$ transform. For the 3-D function $f(x_1, x_2, x_3)$, this is given by

$$\mathcal{Z}_3\{f(x_1,x_2,x_3)\} = \sum_{x_1=-\infty}^{\infty}\sum_{x_2=-\infty}^{\infty}\sum_{x_3=-\infty}^{\infty} f(x_1,x_2,x_3) z_1^{-x_1} z_2^{-x_2} z_3^{-x_3}, \tag{1}$$

where z_1, z_2, and z_3 are complex numbers. Linear time invariant filters are usually classified into recursive and non-recursive. The latter, called finite impulse response (FIR) filters, are very popular due to their simplicity and natural stability. Typically, they are described by a transfer function which is the $\mathcal{Z}$ transform of its impulse response. For the 1-D case, it is given by

$$T(z) = a_0 + a_1 z^{-1} + \cdots + a_n z^{-n} = \sum_{i=0}^{n} a_i z^{-i}, \tag{2}$$

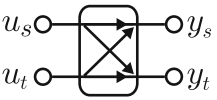
where $a_0, \ldots, a_n$ are real constant coefficients. In the 3-D case, the transfer function of an FIR filter extends to

$$T(z_h, z_v, z_d) = \sum_{i=0}^{n}\sum_{j=0}^{m}\sum_{k=0}^{l} a_{ijk} z_h^{-i} z_v^{-j} z_d^{-k}. \tag{3}$$

In this paper, we deal with a class of orthogonal filters. Introducing the energy of a 3-D real vector function $f(x_1, x_2, x_3)$, in the form

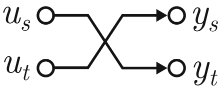
$$\mathcal{E}\{f(x_1,x_2,x_3)\} = \sum_{x_1=-\infty}^{\infty}\sum_{x_2=-\infty}^{\infty}\sum_{x_3=-\infty}^{\infty} f^{\mathrm{T}}(x_1,x_2,x_3)f(x_1,x_2,x_3), \tag{4}$$

we define an orthogonal filter to be a system which preserves the energy, i.e., the input energy equals the output energy. Technically, we are about to find a net of Givens rotations which realizes a given transfer function between two ports of that net. The Givens rotation rotates a vector $[u_s, u_t]^{\mathrm{T}}$ by α radians, i.e., it implements the following set of equations



$$\begin{cases} y_s = u_s\cos(\alpha) - u_t\sin(\alpha) \\ y_t = u_s\sin(\alpha) + u_t\cos(\alpha) \end{cases}. \tag{5}$$

We will denote it by $R_{s,t}(\alpha)$ and use the graphical symbol shown in (5). They are usually implemented using an iterative algorithm called CORDIC [11]. To obtain high throughput, we also utilise permutations in our structures, given by



$$\begin{cases} y_s = u_t \\ y_t = u_s \end{cases}. \tag{6}$$

It is known that the following holds:

Theorem 1 [7] *Transfer function $H(z_1, z_2, z_3)$ of a 3-D orthogonal filter satisfies*

1. *$H(z_1, z_2, z_3)$ is rational in z_1, z_2, z_3;*
2. *$H(z_1, z_2, z_3)$ is holomorfic for all $|z_1| \geq 1, |z_2| \geq 1, |z_3| \geq 1$;*
3. *$H(z_1, z_2, z_3)$ is paraunitary, i.e.*

$$H^{\mathrm{T}}(z_1^{-1}, z_2^{-1}, z_3^{-1})H(z_1, z_2, z_3) = I. \tag{7}$$

The paraunitary condition (7) imposes the single-input single-output system to have a constant, unit frequency response. Typically, orthogonal systems have more inputs and outputs (at least two outputs) to implement practical non-flat frequency responses. During the synthesis process, we use a state space representation [17]. For given

$$u(d) = \begin{bmatrix} u_1(d) \\ u_2(d) \\ \vdots \\ u_{N_u}(d) \end{bmatrix}, \quad y(d) = \begin{bmatrix} y_1(d) \\ y_2(d) \\ \vdots \\ y_{N_y}(d) \end{bmatrix}, \quad x(d) = \begin{bmatrix} x_1(d) \\ x_2(d) \\ \vdots \\ x_{N_x}(d) \end{bmatrix}, \tag{8}$$

which are input, output, and state vector, respectively, it is defined by

$$\begin{bmatrix} x(d+1) \\ y(d) \end{bmatrix} = \tau \begin{bmatrix} x(d) \\ u(d) \end{bmatrix}. \tag{9}$$

τ is a real constant partitioned matrix of the form

$$\tau = \begin{bmatrix} A & B \\ C & D \end{bmatrix}. \tag{10}$$

2-D state space equations, known as the Roesser model, are given by [18]

$$\begin{bmatrix} x_h(h+1, v) \\ x_v(h, v+1) \\ y(h, v) \end{bmatrix} = \tau \begin{bmatrix} x_h(h, v) \\ x_v(h, v) \\ u(h, v) \end{bmatrix}. \tag{11}$$

Similarly, $u(h, v)$, $y(h, v)$ and $x_h(h, v)$, $x_v(h, v)$ are the 2-D input, output, and two state vectors in h and v directions, respectively, and can be represented in the following form

$$\begin{aligned} u(h, v) &= \begin{bmatrix} u_1(h, v) \\ u_2(h, v) \\ \vdots \\ u_{N_u}(h, v) \end{bmatrix}, \quad y(h, v) = \begin{bmatrix} y_1(h, v) \\ y_2(h, v) \\ \vdots \\ y_{N_y}(h, v) \end{bmatrix}, \\ x_h(h, v) &= \begin{bmatrix} x_{h1}(h, v) \\ x_{h2}(h, v) \\ \vdots \\ x_{hN_h}(h, v) \end{bmatrix}, \quad x_v(h, v) = \begin{bmatrix} x_{v1}(h, v) \\ x_{v2}(h, v) \\ \vdots \\ x_{vN_v}(h, v) \end{bmatrix}. \end{aligned} \tag{12}$$

It is easy to show that if τ is an orthogonal matrix, i.e., $\tau^{\mathrm{T}}\tau = I$, then the systems described by (9) and (11) are orthogonal filters. During synthesis, this property allow us to focus on the state space matrices orthogonality instead of the energy of signals.

We say that a system described by the impulse response $r(h, v, d)$ is separable if it can be represented in the form

$$r(h, v, d) = r_h(h)r_v(v)r_d(d). \tag{13}$$

Otherwise, we call it non-separable.

Nowadays, one can observe that most of n-D DSP designs are based on intuitive manipulation applied to a given computational algorithm like ordering multipliers and adders to speed up calculations. Most of the time, no other parameters, except speed and chip area occupation, are taking into account. Unfortunately, when one improves one parameter, another gets worse. This can readily be seen when comparing direct form structures of infinite impulse response 1-D digital filter (fast and inaccurate) and cascade ones (more accurate but output is delayed) for high orders. Similar problems are observed in active electronic filters compared to passive ones made of inductors, capacitors, and resistors, as well as in classic 1-D DSP direct form structures versus wave and orthogonal filters. There are several other parameters influencing the design performance like the sensitivity of the frequency response to changes in the structural parameters, noise, intrinsic oscillations, and limit cycles. So, the idea which motivated the authors in this paper is that one can design better n-D systems by incorporating the techniques which helped in 1-D domain like the orthogonal filters technique. A novelty of our approach is that we can design orthogonal 3-D filters consisting of Givens rotations. To the best of the authors' knowledge, no other techniques to synthesize 3-D rotation structures have yet been published. The scope of the paper is to present details of the synthesis algorithms for separable and non-separable systems. The main idea of our approach is to decompose a 3-D systems into a connection of lower dimension blocks and then apply previously elaborated, 1-D and 2-D orthogonal synthesis procedures [24,26].

The paper is organized as follows. In Sect. 2.1, we present details of the synthesis algorithm of separable orthogonal 3-D filters. Next, an example which illustrates that method is presented in Sect. 2.2 (3-D Gaussian filter). In Sect. 3.1, an expanded version of the technique for the non-separable case is proposed followed by its application in Sect. 3.2 (3-D Laplace filter). In the examples included in this paper, we utilize the standard sample by sample ordering which converts the 3-D signal into 2-D images processed image by image.

2 Separable Orthogonal 3-D FIR Filters

2.1 Synthesis Algorithm

For a separable 3-D system, we represent its transfer function (3) in the form

$$\begin{aligned} T(z_h, z_v, z_d) &= T_h(z_h)T_v(z_v)T_d(z_d) \\ &= \left(\sum_{i=0}^{n} a_{hi} z_h^{-i}\right)\left(\sum_{j=0}^{m} a_{vj} z_v^{-j}\right)\left(\sum_{k=0}^{l} a_{dk} z_d^{-k}\right). \end{aligned} \tag{14}$$

Applying the 1-D synthesis technique, presented in [25], to $T_h(z_h)$, $T_v(z_v)$, and $T_d(z_d)$ separately, we have paraunitary systems $H_h(z_h)$, $H_v(z_v)$, and $H_d(z_d)$ for which we obtain three 1-D state space realizations spanned on a 3-D domain:

$$\begin{bmatrix} x_h(h+1, v, d) \\ y_h(h, v, d) \end{bmatrix} = \tau_h \begin{bmatrix} x_h(h, v, d) \\ u(h, v, d) \end{bmatrix} \text{ for } H_h(z_h) \tag{15a}$$

$$\begin{bmatrix} x_v(h, v+1, d) \\ y_v(h, v, d) \end{bmatrix} = \tau_v \begin{bmatrix} x_v(h, v, d) \\ u_v(h, v, d) \end{bmatrix} \text{ for } H_v(z_v) \tag{15b}$$

$$\begin{bmatrix} x_d(h, v, d+1) \\ y_d(h, v, d) \end{bmatrix} = \tau_d \begin{bmatrix} x_d(h, v, d) \\ u_d(h, v, d) \end{bmatrix} \text{ for } H_d(z_d). \tag{15c}$$

We also get scaling factors k_h, k_v and k_d, which are the maxima of the frequency response squared for $H_h(z_h)$, $H_v(z_v)$, and $H_d(z_d)$, respectively. Then, each state space realization in (15) is implemented using pipeline structures H_h, H_v, and H_d, respectively. Constructing a cascade connection of these three implementations, we obtain the separable 3-D orthogonal pipeline structure shown in Fig. 1.

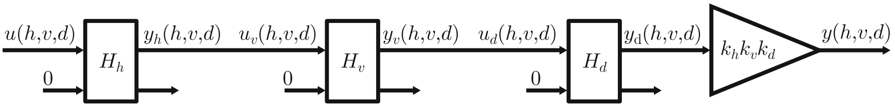


Fig. 1 Block diagram of a 3-D separable orthogonal FIR filter

2.2 Realization Example of a Gaussian Filter

Let us give a design example of a separable 3-D FIR Filter. We have chosen a Gaussian filter whose impulse response is given by

$$r(h, v, d) = q^3 \exp\left(-\frac{h^2 + v^2 + d^2}{2\sigma^2}\right), \tag{16}$$

where

$$q = \sqrt[3]{\frac{1}{\sigma\sqrt{2\pi}}}. \tag{17}$$

Substituting (16) into (13), we have

$$r_h(h) = r(h), \quad r_v(v) = r(v), \quad r_d(d) = r(d), \tag{18}$$

where

$$r(x) = q \cdot \exp\left(-\frac{x^2}{2\sigma^2}\right). \tag{19}$$

By (14), we obtain

$$T_h(z_h) = T(z_h), \quad T_v(z_v) = T(z_v), \quad T_d(z_d) = T(z_d), \tag{20}$$

where

$$T(z) = \sum_{i=0}^{N} r(i - n_0) z^{-i}. \tag{21}$$

The constant n_0 is chosen to shift the origin of (19). The parameters for the filter have been taken from [1], i.e., $\sigma = 0.7$, $n_0 = 2$, and $h, v, d = -2, -1, 0, 1, 2$. Substituting them into (21), we obtain kernel coefficients of the Gaussian filter which are presented in Table 1. From (20) we see that it suffice to design a state space system by the technique presented in [25] applied to $T(z)$, which is given by

$$\begin{aligned} T(z) = 0.0139950 + 0.2988453z^{-1} + 0.8290945z^{-2} \\ + 0.2988453z^{-3} + 0.0139950z^{-4}. \end{aligned} \tag{22}$$

Table 1 Kernel coefficients of 3-D Gaussian filter ($\sigma = 0.7$)

$d = -2 = 2$	$h = -2 = 2$	$h = -1 = 1$	$h = 0$
$v = -2 = 2$	0.0000027	0.0000585	0.0001624
$v = -1 = 1$	0.0000585	0.0012499	0.0034676
$v = 0$	0.0001624	0.0034676	0.0096201
$d = -1 = 1$	$h = -2 = 2$	$h = -1 = 1$	$h = 0$
$v = -2 = 2$	0.0000585	0.0012499	0.0034676
$v = -1 = 1$	0.0012499	0.0266894	0.0740452
$v = 0$	0.0034676	0.0740452	0.2054255
$d = 0$	$h = -2 = 2$	$h = -1 = 1$	$h = 0$
$v = -2 = 2$	0.0001624	0.0034676	0.0096201
$v = -1 = 1$	0.0034676	0.0740452	0.2054255
$v = 0$	0.0096201	0.2054255	0.5699175

The synthesis algorithm is as follows. We construct a paraunitary transfer vector:

$$H(z) = \begin{bmatrix} \frac{1}{\sqrt{k}} T(z) \\ G(z) \end{bmatrix}$$

$$= \begin{bmatrix} 0.0096201 + 0.2054237z^{-1} + 0.5699125z^{-2} + 0.2054237z^{-3} + 0.0096201z^{-4} \\ 0.5882043 - 0.4851375z^{-1} - 0.0960584z^{-2} - 0.0068492z^{-3} - 0.0001573z^{-4} \end{bmatrix} \tag{23}$$

where $G(z^{-1})G(z) = 1 - \frac{1}{k} T^{\mathrm{T}}(z^{-1})T(z)$ and k is a constant scalar chosen to make the factorization possible. We have chosen $k = k_h = k_v = k_d = 1.4547750$ which is the maximum of the frequency response of $|T(e^{j\omega})|^2$. Next, we represent (23) in the form:

$$H(z) = \sum_{i=0}^{m} H_i z^{-i} \tag{24}$$

where:

$$H_0 = \begin{bmatrix} 0.0096201 \\ 0.5882043 \end{bmatrix} H_1 = \begin{bmatrix} 0.2054237 \\ -0.4851375 \end{bmatrix} H_2 = \begin{bmatrix} 0.5699125 \\ -0.0960584 \end{bmatrix}$$

$$H_3 = \begin{bmatrix} 0.2054237 \\ -0.0068492 \end{bmatrix} H_4 = \begin{bmatrix} 0.0096201 \\ -0.0001573 \end{bmatrix}. \tag{25}$$

By applying QR to $\begin{bmatrix} H_1 & H_2 & H_3 & H_4 \\ H_2 & H_3 & H_4 & 0 \\ H_3 & H_4 & 0 & 0 \\ H_4 & 0 & 0 & 0 \end{bmatrix} = \mathrm{QR} = Q \begin{bmatrix} M \\ 0 \end{bmatrix}$, we find the full rank $M = \begin{bmatrix} M_0 & M_1 & M_2 & M_3 \end{bmatrix}$, where

$$M_0 = \begin{bmatrix} -0.8086551 \\ 0 \\ 0 \\ 0 \end{bmatrix} M_1 = \begin{bmatrix} -0.3504384 \\ -0.5035462 \\ 0 \\ 0 \end{bmatrix}$$
$$M_2 = \begin{bmatrix} -0.0630916 \\ -0.1938233 \\ 0.0281133 \\ 0 \end{bmatrix} M_3 = \begin{bmatrix} -0.0025382 \\ -0.0091515 \\ 0.0015418 \\ -0.0000022 \end{bmatrix}. \tag{26}$$

For (23), we determine the state space representation (9), where $A = \begin{bmatrix} M_1 & M_2 & M_3 & 0 \end{bmatrix} M^{-1}$, $B = M_0$, $C = \begin{bmatrix} H_1 & H_2 & H_3 & H_4 \end{bmatrix} M^{-1}$, and $\hat{D} = H_0$:

$$A = \begin{bmatrix} 0.4333595 & -0.1762979 & -0.3332045 & -0.1269695 \\ 0.6226959 & -0.0484429 & 0.7379438 & 0.2552332 \\ 0 & -0.0558305 & -0.3300750 & 0.9408855 \\ 0 & 0 & -0.0000769 & -0.0548415 \end{bmatrix}$$
$$B = \begin{bmatrix} -0.8086551 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$
$$C = \begin{bmatrix} -0.2540313 & -0.9550070 & 0.1527338 & -0.0030788 \\ 0.5999313 & -0.2267528 & -0.4605829 & -0.1745056 \end{bmatrix}$$
$$\hat{D} = \begin{bmatrix} 0.0096201 \\ 0.5882043 \end{bmatrix}. \tag{27}$$

We apply similarity transformation to (27) starting from Schur decomposition of A (to find upper triangular representation) which changes also B and C which leads to a new matrices A_U, $\hat{B}_U$, and C_U

$$A_U = \begin{bmatrix} 0 & -0.9400852 & 0.2544060 & 0.0773702 \\ 0 & 0 & 0.6657319 & -0.2543591 \\ 0 & 0 & 0 & 0.9401080 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$
$$\hat{B}_U = \begin{bmatrix} -0.2133518 \\ 0.7014064 \\ 0.3408316 \\ 0.0163420 \end{bmatrix}$$
$$C_U = \begin{bmatrix} -0.9998661 & -0.0056070 & -0.0114712 & -0.0034886 \\ 0.0163636 & -0.3408934 & -0.7013894 & -0.2133073 \end{bmatrix}. \tag{28}$$

Next, we extend $\begin{bmatrix} A_U & \hat{B}_U \\ C_U & \hat{D} \end{bmatrix}$ to be a square matrix [27,29] which leads to final state space realization, given by

$$\begin{aligned}
A_h = A_v = A_d &= \begin{bmatrix} 0 & -0.9400852 & 0.2544060 & 0.0773702 \\ 0 & 0 & 0.6657319 & -0.2543591 \\ 0 & 0 & 0 & 0.9401080 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
B_h = B_v = B_d &= \begin{bmatrix} -0.2133518 & 0.0034871 \\ 0.7014064 & -0.0114639 \\ 0.3408316 & -0.0055706 \\ 0.0163420 & 0.9998665 \end{bmatrix} \\
C_h = C_v = C_d &= \begin{bmatrix} -0.9998661 & -0.0056070 & -0.0114712 & -0.0034886 \\ 0.0163636 & -0.3408934 & -0.7013894 & -0.2133073 \end{bmatrix} \\
D_h = D_v = D_d &= \begin{bmatrix} 0.0096201 & -0.0001573 \\ 0.5882043 & -0.0096137 \end{bmatrix}.
\end{aligned} \tag{29}$$

Applying the Givens decomposition and permutations to (29), we obtain the pipeline rotation structure shown in Fig. 2 whose parameters are presented in Table 2 (detailed algorithm of the decomposition is presented in [26]).

To apply the obtained 3-D Gaussian filter to image processing, we need to link each independent variable h, v, and d with the dimensions of an image. To keep things simple, we assign h, v, and d to be indexes of the 3-D image in the horizontal (rows), vertical (columns), and depth (frames) dimensions, respectively. Suppose that an image frame is of size $H \times V$. We assume that it is processed sample by sample

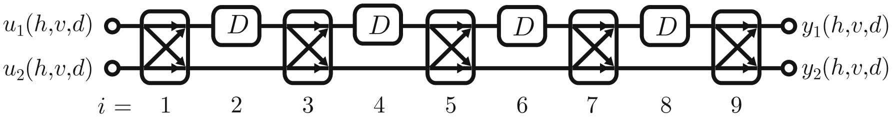


Fig. 2 Block diagram of the 1-D sub-block of the 3-D orthogonal Gaussian filter

Table 2 Component parameters of the 1-D sub-block of the 3-D Gaussian filter

i	Component	α
1	Rotation	−1.5544512
2	Permutation, delay	–
3	Rotation	−2.793736
4	Permutation, delay	–
5	Rotation	0.8423221
6	Permutation, delay	–
7	Rotation	−2.793684
8	Permutation, delay	–
9	Rotation	0.0163619

using row by row and frame by frame ordering. To do this, the unit delays D in each state space system H_h, H_v, and H_d, shown in Fig. 2, need to be replaced with the following 1-D delays: z^{-1}, z^{-H}, and z^{-HV}, respectively. The 3-D Gaussian filter has been modelled in the Scilab environment [22]. The impulse response of the filter has been simulated for the $8 \times 8 \times 8$ 3-D Kronecker delta matrix input. Obtained results are similar to shown in Table 1 with mean-squared error which equals to $6.346 \cdot 10^{-23}$. The filter has been tested with a real 3-D medical DICOM image taken from [4]. For this task, the authors have implemented Scilab procedures to read and write DICOM files.

3 Non-separable Orthogonal 3-D FIR Filters

3.1 Synthesis Algorithm

Suppose we are given a non-separable transfer function (3). To obtain a rotation structure in this case, we rearrange (3) into one of the following form:

$$H(z_h, z_v, z_d) = \sum_{j=0}^{m} H_j(z_h, z_d) z_v^{-j} \tag{30a}$$

$$= \sum_{i=0}^{n} H_i(z_v, z_d) z_h^{-i} \tag{30b}$$

$$= \sum_{k=0}^{l} H_k(z_v, z_h) z_d^{-k}. \tag{30c}$$

From (30), we have three ways to decompose the system into a cascade connection of 2-D and 1-D blocks. We have chosen (30a); the other two structures can be obtained in a similar way. The coefficients $H_j(z_h, z_d)$ can be represented in the so-called Gram form

$$H_j(z_h, z_d) = Z_d C_j Z_h, \tag{31}$$

where

$$C_j = \begin{bmatrix} a_{0j0} & a_{1j0} & \cdots & a_{nj0} \\ a_{0j1} & a_{1j1} & \cdots & a_{nj1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{0jl} & a_{1jl} & \cdots & a_{njl} \end{bmatrix}, \tag{32}$$

and $Z_d = \left[1 \; z_d^{-1} \cdots z_d^{-l}\right]$, $Z_h = \left[1 \; z_h^{-1} \cdots z_h^{-n}\right]^{\mathrm{T}}$. Introducing the partitioned matrix

$$C = \left[C_0 \; C_1 \cdots C_m\right], \tag{33}$$

we can represent (30a) in the form

$$H(z_h, z_v, z_d) = Z_d C Z_e, \tag{34}$$

where

$$Z_e = \begin{bmatrix} Z_h \cdot z_v^0 \\ Z_h \cdot z_v^{-1} \\ \vdots \\ Z_h \cdot z_v^{-m} \end{bmatrix}. \tag{35}$$

Applying a full rank factorization [8] to (33), we obtain

$$C = C_d C_e, \tag{36}$$

where C_d and C_e are real full rank $(l+1) \times r$ and $r \times (n+1)$ matrices, respectively. Substituting (36) into (34), we obtain

$$H(z_h, z_v, z_d) = \hat{H}_d(z_d) H_e(z_h, z_v), \tag{37}$$

where

$$\hat{H}_d(z_d) = Z_d C_d = \left[H_{d_1}(z_d)\ H_{d_2}(z_d)\ \cdots\ H_{d_r}(z_d) \right], \tag{38}$$

and

$$H_e(z_h, z_v) = C_e Z_e = \begin{bmatrix} H_{e_1}(z_h, z_v) \\ H_{e_2}(z_h, z_v) \\ \vdots \\ H_{e_r}(z_h, z_v) \end{bmatrix}. \tag{39}$$

Each $H_{e_i}(z_h, z_v)$ is an single-input single-output system, so we can apply the algorithm presented in [27,28] to synthesize 2-D orthogonal state space equations, given by (11), spanned on 3-D. The factors k_{e_i} obtained from the 2-D synthesis are applied to the corresponding elements of $\hat{H}_d(z_d)$. The resulting vector will be denoted by

$$H_d(z_d) = \left[k_{e_1} H_{d_1}(z_d)\ k_{e_2} H_{d_2}(z_d)\ \cdots\ k_{e_r} H_{d_r}(z_d) \right]. \tag{40}$$

Thanks to it, we remove r scaling factors from the rotation structure. As (40) is a horizontal vector, it cannot satisfy (7). So, to obtain an orthogonal structure, we take into account a transposed version of (40) [17, p. 546]. So, we can apply the algorithm illustrated in Sect. 2.2 to $H_d^{\mathrm{T}}(z_d)$ obtaining (10) which is a square matrix [27]. Then, the result is transposed back to the orthogonal realization of (40), which is 1-D state space equations (9) spanned on 3-D (dimensions h and v are not processed), given by

$$\begin{bmatrix} x_d(h, v, d+1) \\ y_d(h, v, d) \end{bmatrix} = \tau_d \begin{bmatrix} x_d(h, v, d) \\ u_d(h, v, d) \end{bmatrix}, \tag{41}$$

where

$$\tau_d = \begin{bmatrix} A_d & B_d \\ C_d & D_d \end{bmatrix} = \begin{bmatrix} A^{\mathrm{T}} & C^{\mathrm{T}} \\ B^{\mathrm{T}} & D^{\mathrm{T}} \end{bmatrix}. \tag{42}$$

In (37), we have a dot product of vectors (38) and (39). From a system point of view, (39) can be represented as r one-input one-output parallel blocks. Equation (38) can be

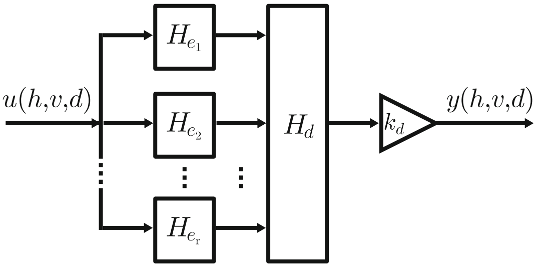


Fig. 3 Block diagram of a 3-D non-separable orthogonal FIR filter decomposed into a cascade of 2-D and 1-D systems

treated as a single system consisting of r inputs and one output. So, the transfer function (37) can be realized as a cascade connection of r $H_{e_i}(z_h, z_v)$ blocks whose outputs are connected to r inputs of $H_d(z_d)$ and one multiplier, as shown in Fig. 3. Symbols H_{e_i} and H_d denote realizations of $H_{e_i}(z_h, z_v)$ and $H_d(z_d)$, respectively. Systems H_{e_i} and H_d can be implemented using Givens rotations (5) and permutations (6) by the technique presented in [26].

3.2 Realization Example of a 3-D Laplace Filter

Let us design the 3-D orthogonal Laplace filter whose mask is given by [12]

$$\left[\begin{bmatrix}0&0&0\\0&1&0\\0&0&0\end{bmatrix}, \begin{bmatrix}0&1&0\\1&-6&1\\0&1&0\end{bmatrix}, \begin{bmatrix}0&0&0\\0&1&0\\0&0&0\end{bmatrix}\right]. \tag{43}$$

The transfer function for the system described by (43) is

$$H(z_h, z_v, z_d) = \begin{bmatrix} z_d^0 & z_d^{-1} & z_d^{-2} \end{bmatrix} C \begin{bmatrix} Z_h z_v^0 \\ Z_h z_v^{-1} \\ Z_h z_v^{-2} \end{bmatrix}, \tag{44}$$

where

$$C = \begin{bmatrix} C_0 & C_1 & C_2 \end{bmatrix} = \begin{bmatrix} 0&0&0&0&1&0&0&0&0 \\ 0&1&0&1&-6&1&0&1&0 \\ 0&0&0&0&1&0&0&0&0 \end{bmatrix} \tag{45}$$

and

$$Z_h = \begin{bmatrix} z_h^0 & z_h^{-1} & z_h^{-2} \end{bmatrix}^{\mathrm{T}}. \tag{46}$$

Applying a full rank factorization to (45), we obtain (37), where $r = 2$,

$$\hat{H}_d(z_d) = \begin{bmatrix} z_d^0 & z_d^{-1} & z_d^{-2} \end{bmatrix} \begin{bmatrix} -0.3748367 & -0.4573964 \\ 2.4869572 & -0.1378785 \\ -0.3748367 & -0.4573964 \end{bmatrix}, \tag{47}$$

and

$$H_e(z_h,z_v)=\begin{bmatrix}H_{e_1}(z_h,z_v)\\H_{e_2}(z_h,z_v)\end{bmatrix}$$
$$=\begin{bmatrix}0 & 0.3846230 & 0 & 0.3846230 & -2.4236794 & 0.3846230 & 0 & 0.3846230 & 0\\0 & -0.3151988 & 0 & -0.3151988 & -0.2000804 & -0.3151988 & 0 & -0.3151988 & 0\end{bmatrix}\begin{bmatrix}Z_h z_v^0\\Z_h z_v^{-1}\\Z_h z_v^{-2}\end{bmatrix}. \tag{48}$$

Let us focus on $H_{e_1}(z_h,z_v)$ realization, given by (48), for a moment. We represent $H_{e_1}(z_h,z_v)$ in the form:

$$H_{e_1}(z_h,z_v)=\begin{bmatrix}z_h^0 & z_h^{-1} & z_h^{-2}\end{bmatrix}P\begin{bmatrix}z_v^0\\z_v^{-1}\\z_v^{-2}\end{bmatrix} \tag{49}$$

where

$$P=\begin{bmatrix}0 & 0.3846230 & 0\\0.3846230 & -2.4236794 & 0.3846230\\0 & 0.3846230 & 0\end{bmatrix}. \tag{50}$$

Applying the full rank decomposition to (50), we obtain:

$$H_{e_1}(z_h,z_v)=H_h^{e_1}(z_h)H_v^{e_1}(z_v) \tag{51}$$

where

$$\begin{aligned}H_h^{e_1}(z_h)&=\begin{bmatrix}z_h^0 & z_h^{-1} & z_h^{-2}\end{bmatrix}\begin{bmatrix}0.2359769 & 0.2359769\\-1.5584563 & 0.0714618\\0.2359769 & 0.2359769\end{bmatrix},\\H_v^{e_1}(z_v)&=\begin{bmatrix}-0.2359769 & 1.5584563 & -0.2359769\\0.2359769 & 0.0714618 & 0.2359769\end{bmatrix}\begin{bmatrix}z_v^0\\z_v^{-1}\\z_v^{-2}\end{bmatrix}.\end{aligned} \tag{52}$$

For (52), we construct the paraunitary systems:

$$\begin{aligned}\left(U_h^{e_1}\right)^{\mathrm{T}}(z_h)&=\begin{bmatrix}\dfrac{1}{\sqrt{k_h^{e_1}}}\left(H_h^{e_1}\right)^{\mathrm{T}}(z_h)\\0\\F_h^{e_1}(z_h)\end{bmatrix}\\&=\begin{bmatrix}0.1140244z_h^0-0.7530483z_h^{-1}+0.1140244z_h^{-2}\\0.1140244z_h^0+0.0345305z_h^{-1}+0.1140244z_h^{-2}\\0\\0.4611571z_h^0+0.4048143z_h^{-1}-0.0563866z_h^{-2}\end{bmatrix}\\U_v^{e_1}(z_v)&=\begin{bmatrix}\dfrac{1}{\sqrt{k_v^{e_1}}}H_v^{e_1}(z_v)\\F_v^{e_1}(z_v)\\0\end{bmatrix}\end{aligned}$$

$$= \begin{bmatrix} -0.1140244 z_v^0 + 0.7530483 z_v^{-1} - 0.1140244 z_v^{-2} \\ 0.1140244 z_v^0 + 0.0345305 z_v^{-1} + 0.1140244 z_v^{-2} \\ 0.4611571 z_v^0 + 0.4048143 z_v^{-1} - 0.0563866 z_v^{-2} \\ 0 \end{bmatrix} \tag{53}$$

We obtain 1-D state space realizations of (53) [in similar way as (24)–(28) in Sect. 2.2]:

$$\begin{aligned} A_t &= \begin{bmatrix} -0.1375940 & 0.1359287 \\ -0.1392797 & 0.1375940 \end{bmatrix} & \hat{B}_t &= \begin{bmatrix} 0.8725430 \\ 0 \end{bmatrix} \\ C_t &= \begin{bmatrix} -0.8630501 & -0.0856547 \\ 0.0395745 & -0.9773549 \\ 0 & 0 \\ 0.4639477 & 0.0056496 \end{bmatrix} & \hat{D}_t &= \begin{bmatrix} 0.1140244 \\ 0.1140244 \\ 0 \\ 0.4611571 \end{bmatrix} \end{aligned} \tag{54}$$

$$\begin{aligned} A_v^{e_1} &= \begin{bmatrix} 0 & 0.2752085 \\ 0 & 0 \end{bmatrix} & B_v^{e_1} &= \begin{bmatrix} 0.6132134 \\ -0.6207259 \end{bmatrix} \\ C_v^{e_1} &= \begin{bmatrix} 0.6674764 & -0.5537755 \\ -0.6674764 & -0.7150272 \\ 0.3300763 & -0.3260813 \\ 0 & 0 \end{bmatrix} & D_v^{e_1} &= \begin{bmatrix} -0.1140244 \\ 0.1140244 \\ 0.4611571 \\ 0 \end{bmatrix} \end{aligned} \tag{55}$$

We add new columns to $\hat{B}_t$ and $\hat{D}_t$, obtaining square matrix $\begin{bmatrix} A_t & B_t \\ C_t & D_t \end{bmatrix}$ [29]:

$$\begin{aligned} B_t &= \begin{bmatrix} 0.8725430 & 0 & 0 & -0.4486204 \\ 0 & 0.9770079 & 0 & 0.0844077 \end{bmatrix} \\ D_t &= \begin{bmatrix} 0.1140244 & -0.1507574 & 0 & 0.4605199 \\ 0.1140244 & 0.1507574 & 0 & -0.0864978 \\ 0 & 0 & 1 & 0. \\ 0.4611571 & 0 & 0 & 0.7563430 \end{bmatrix}. \end{aligned} \tag{56}$$

Based on (56), we construct a transpose system which is the realization of $U_h^{e_1}$, given by

$$\begin{aligned} A &= \begin{bmatrix} -0.1375940 & -0.1392797 \\ 0.1359287 & 0.1375940 \end{bmatrix} & B &= \begin{bmatrix} -0.8630501 & 0.0395745 & 0 & 0.4639477 \\ -0.0856547 & -0.9773549 & 0 & 0.0056496 \end{bmatrix} \\ C &= \begin{bmatrix} 0.8725430 & 0 \\ 0 & 0.9770079 \\ 0 & 0 \\ -0.4486204 & 0.0844077 \end{bmatrix} & D &= \begin{bmatrix} 0.1140244 & 0.1140244 & 0 & 0.4611571 \\ -0.1507574 & 0.1507574 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0.4605199 & -0.0864978 & 0 & 0.7563430 \end{bmatrix} \end{aligned} \tag{57}$$

We apply Schur upper triangularization to matrix A, obtaining a minimized state space model for $U_h^{e_1}$:

$$
\begin{aligned}
A_h^{e_1} &= \begin{bmatrix} 0 & 0.2752085 \\ 0 & 0 \end{bmatrix} \\
B_h^{e_1} &= \begin{bmatrix} 0.5537755 & -0.7150272 & 0 & -0.3260813 \\ -0.6674764 & -0.6674764 & 0 & 0.3300763 \end{bmatrix} \\
C_h^{e_1} &= \begin{bmatrix} -0.6207259 & 0.6132134 \\ 0.6866301 & 0.6950421 \\ 0 & 0 \\ 0.3784687 & -0.2552378 \end{bmatrix} \\
D_h^{e_1} &= \begin{bmatrix} 0.1140244 & 0.1140244 & 0 & 0.4611571 \\ -0.1507574 & 0.1507574 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0.4605199 & -0.0864978 & 0 & 0.7563430 \end{bmatrix}
\end{aligned}
\tag{58}
$$

Then, we use 1-D state space systems (58) and (55) to construct 2-D state space model:

$$
\begin{aligned}
A_R^{e_1} &= \begin{bmatrix} A_h^{e_1} & \left(C_h^{e_1}\right)^{\mathrm{T}} C_v^{e_1} \\ 0_{h\times v} & \left(A_v^{e_1}\right)^{\mathrm{T}} \end{bmatrix} \\
&= \begin{bmatrix} 0 & 0.2752085 & 0.8468960 & 0.2045966 \\ 0 & 0 & 0 & 0.8468960 \\ 0 & 0 & 0 & 0.2752085 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
C_R^{e_1} &= \begin{bmatrix} \left(B_h^{e_1}\right)^{\mathrm{T}} & \left(D_h^{e_1}\right)^{\mathrm{T}} C_v^{e_1} \end{bmatrix} \\
&= \begin{bmatrix} -0.6207259 & 0.6132134 & 0 & -0.1446744 \\ 0.6866301 & 0.6950421 & -0.2012541 & -0.0243099 \\ 0 & 0 & 0.3300763 & -0.3260813 \\ 0.3784687 & -0.2552378 & 0.3651214 & -0.1931763 \end{bmatrix}
\end{aligned}
\tag{59}
$$

$$
\begin{aligned}
\hat{B}_R^{e_1} &= \begin{bmatrix} \left(C_h^{e_1}\right)^{\mathrm{T}} D_v^{e_1} \\ B_v^{e_1} \end{bmatrix} = \begin{bmatrix} -0.1446744 \\ 0 \\ 0.6132134 \\ -0.6207259 \end{bmatrix} \\
\hat{D}_R^{e_1} &= \left(D_h^{e_1}\right)^{\mathrm{T}} D_v^{e_1} = \begin{bmatrix} 0 \\ 0.0343800 \\ 0.4611571 \\ -0.0623733 \end{bmatrix}
\end{aligned}
\tag{60}
$$

We extend the number of columns in (60):

$$B_R^{e_1} = \begin{bmatrix} -0.1446744 & 0 & -0.2716726 & -0.2653886 \\ 0 & -0.2252368 & 0.4343535 & 0.2082612 \\ 0.6132134 & 0.6931186 & -0.2329707 & 0.1163662 \\ -0.6207259 & 0.6847299 & 0.3787022 & -0.0492859 \end{bmatrix}$$

$$D_R^{e_1} = \begin{bmatrix} 0 & 0 & 0.0602207 & 0.4627221 \\ 0.0343800 & 0 & 0.0544406 & -0.0146858 \\ 0.4611571 & 0 & 0.7302397 & -0.1969889 \\ -0.0623733 & 0 & 0 & 0.7855533 \end{bmatrix}. \tag{61}$$

Finally, we apply Givens decomposition and permutations to (59) and (61), obtaining the pipeline rotation structure.

Repeating steps (49)–(61) to $H_{e_2}(z_h, z_v)$ in (48), we design a 2-D state space model:

$$A_R^{e_2} = \begin{bmatrix} 0 & 0.6204384 & 0.6160282 & -0.2064796 \\ 0 & 0 & 0 & 0.6160282 \\ 0 & 0 & 0 & 0.6204384 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B_R^{e_2} = \begin{bmatrix} -0.2723030 & -0.1275783 & -0.2764642 & 0.1614744 \\ 0 & 0 & 0 & 0.7877241 \\ -0.2596936 & -0.4926194 & -0.2636621 & -0.4852049 \\ -0.7124480 & 0 & 0.7017249 & 0 \end{bmatrix}$$

$$C_R^{e_2} = \begin{bmatrix} 0.7124480 & 0.2596936 & 0 & 0.2723030 \\ -0.5610438 & -0.2045055 & 0.3320120 & 0.2345038 \\ 0 & 0 & 0.5612125 & 0.2045669 \\ -0.4214827 & 0.7111912 & -0.4419477 & 0.1481315 \end{bmatrix}$$

$$D_R^{e_2} = \begin{bmatrix} 0 & 0.5527025 & 0 & -0.2129507 \\ -0.1467593 & 0.6330967 & -0.1490020 & -0.1833903 \\ 0.5395614 & -0.1624233 & 0.5478065 & -0.1599786 \\ 0.1953542 & 0.0915265 & 0.1983395 & -0.1158441 \end{bmatrix}, \tag{62}$$

for which we can also get the pipeline rotation structure. In a similar way as in Sect. 2.2, (47) can by realized by (41), where

$$A^d = \begin{bmatrix} 0 & 0.3132932 \\ 0 & 0 \end{bmatrix} B^d = \begin{bmatrix} 0.2089567 & -0.7822908 & 0.4961910 \\ -0.8738511 & -0.4045045 & -0.2697411 \end{bmatrix}$$

$$C^d = \begin{bmatrix} 0.4212768 & -0.7749423 \\ 0.8635104 & 0.4694245 \\ -0.2772646 & 0.2845217 \end{bmatrix} D^d = \begin{bmatrix} -0.1153337 & -0.0533878 & 0.4536941 \\ -0.0774472 & 0.1673091 & 0 \\ -0.4164389 & 0.4399483 & 0.6893445 \end{bmatrix} \tag{63}$$

The state space equations (59), (61), (62), and (63) are implementations of blocks given in Fig. 3. Replacing H_{e_1}, H_{e_2}, and H_d with their orthogonal counterparts, we

Table 3 Impulse response of the 3-D Laplace filter

$d=-1$	$h=-1$	$h=0$	$h=1$
$v=-1$	0	0.0000006	0
$v=0$	0.0000006	0.9999989	0.0000004
$v=1$	0	0.0000004	0

$d=0$	$h=-1$	$h=0$	$h=1$
$v=-1$	0.0000005	0.9999985	0.0000003
$v=0$	0.9999986	−5.9999964	0.9999996
$v=1$	0.0000002	0.9999997	0

$d=1$	$h=-1$	$h=0$	$h=1$
$v=-1$	0	0.0000002	0
$v=0$	0.0000002	0.9999995	0
$v=1$	0	0	0

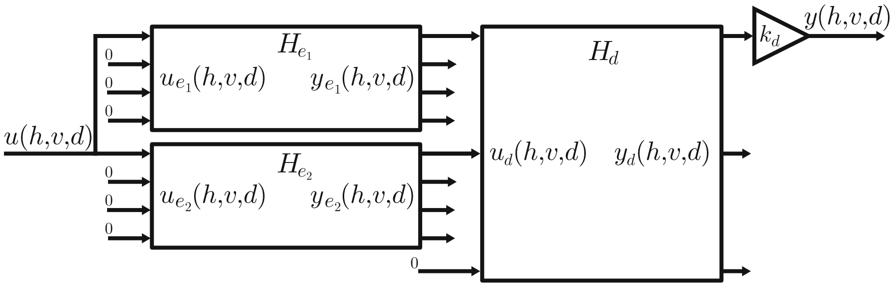


Fig. 4 Pipeline implementation of the 3-D orthogonal Laplace filter

obtain the pipeline structure shown in Fig. 4. Each H_{e_1}, H_{e_2}, and H_d are implemented using Givens rotations and permutations whose parameters are presented in Tables 4, 5, and 6. As they are multi-input multi-output orthogonal systems, there are extra inputs which will be set to zero as well as additional outputs will not be used. Hence, the first three rotations R_1, R_2, and R_3, and the -1 multiplier in the implementation of H_{e_1} and rotations R_{19}, R_{20}, and R_{21}, and the -1 multiplier in the implementation of H_{e_2}, have input and output constantly equal to zero. So, they can be removed from the structure. The final structure for the 3-D orthogonal Laplace filter is shown in Fig. 5. If the H_{e_i} systems have unequal numbers of rotations, we apply extra delay elements to compensate for the processing time of all the H_{e_i} blocks.

The impulse response of the filter has been simulated for the $8\times8\times8$ 3-D Kronecker delta matrix input. The results are presented in Table 3.

4 Conclusions

The main contribution of the paper is the extension of 1-D and 2-D FIR orthogonal filters to 3-D case. By doing so, we open up new possibilities to take into account

Table 4 Rotation and permutation parameters for the H_{e_1} system

R_i/D_i	s_i	t_i	Component	$\cos(\alpha_i)$	$\sin(\alpha_i)$
R_1	7	8	Rotation	−0.2604492	−0.9654876
R_2	6	7	Rotation	0.9729109	−0.2311803
R_3	6	8	Rotation	0	1
R_4	5	6	Rotation	0.8538100	0.5205847
R_5	5	7	Rotation	0.9614384	−0.2750203
R_6	5	8	Rotation	−0.7561659	−0.6543800
D_1	4	5	Permutation, delay	–	–
R_7	5	6	Rotation	0.9384402	−0.3454419
R_8	5	7	Rotation	0.6080205	0.7939213
R_9	5	8	Rotation	0.4823219	0.8759941
D_2	3	5	Permutation, delay	–	–
R_{10}	5	6	Rotation	0.3300763	−0.9439542
R_{11}	5	7	Rotation	0	1.0000000
R_{12}	5	8	Rotation	−0.6000000	−0.8000000
D_3	2	5	Permutation, delay	–	–
R_{13}	5	6	Rotation	0.2932616	0.9560322
R_{14}	5	8	Rotation	0.9384401	−0.3454419
D_4	1	5	Permutation, delay	–	–
R_{15}	5	6	Rotation	0.7270070	0.6866301
R_{16}	5	8	Rotation	−0.8538101	0.5205847

parameters which are usually omitted in 3-D designs like sensitivity of the frequency response to changes in structural parameters, noise, intrinsic oscillations, and limit cycles. This is the main difference from 3-D FIR techniques, known in literature, which usually focus on speed and chip area only. However, presented results are occupied by much higher mathematical burden during synthesis which calls for polynomial factorization, matrix pseudoinverse, QR decomposition, full rank matrix factorization, real Schur decomposition, and orthonormal basis extension. Nonetheless, these are standard numerical methods which can be implemented using popular mathematical software. The authors have applied Scilab [22] for these tasks.

The frequency response of a separable system is limited to a superposition of, possibly different, 1-D functions applied to each direction independently. Due to that, a separable system cannot fully exploit a neighbourhood of a processed sample. On the other hand, their synthesis techniques in n-D are as simple as 1-D approaches applied n times. In the non-separable case, we are allowed to design systems which approximate any frequency response at a cost of much more complicated synthesis algorithms. The originality of our non-separable technique, presented in Sect. 3.1, is that we represent a 3-D system in the Gram form with coefficients collected in a matrix which is a subject of a full rank factorization. Thanks to it, we separate one variable from the system at the expense of an increase in the subsystem's inputs and outputs. As

Table 5 Component parameters for the H_{e_2} system

R_i/D_i	s_i	t_i	Component	$\cos(\alpha_i)$	$\sin(\alpha_i)$
R_{17}	7	8	Rotation	0.9129242	−0.4081291
R_{18}	6	7	Rotation	0.9953631	0.0961889
R_{19}	6	8	Rotation	0.9412373	−0.3377459
R_{20}	5	6	Rotation	0.9772196	−0.2122309
R_{21}	5	7	Rotation	0.7527164	0.6583449
R_{22}	5	8	Rotation	−0.9685670	0.2487529
D_5	4	5	Permutation, delay	–	–
R_{23}	5	6	Rotation	0.9470266	0.3211552
R_{24}	5	7	Rotation	0.9232259	0.3842575
R_{25}	5	8	Rotation	0.7096244	0.7045802
D_6	3	5	Permutation, delay	–	–
R_{26}	5	6	Rotation	0.9103157	0.4139147
R_{27}	5	7	Rotation	0.7873524	0.6165031
R_{28}	5	8	Rotation	0	−1.0000000
D_7	2	5	Permutation, delay	–	–
R_{29}	5	6	Rotation	0.9690025	−0.2470511
R_{30}	5	8	Rotation	0.6402857	0.7681368
D_8	1	5	Permutation, delay	–	–
R_{31}	5	6	Rotation	0.8277861	−0.5610438
R_{32}	5	8	Rotation	0.8606668	−0.5091686

Table 6 Component parameters for the H_d system

R_i/D_i	s_i	t_i	Component	$\cos(\alpha_i)$	$\sin(\alpha_i)$
R_{33}	3	4	Rotation	0.9074892	−0.4200755
R_{34}	3	5	Rotation	−0.9629329	−0.2697412
D_9	2	3	Permutation, delay	–	–
R_{35}	3	4	Rotation	0.3655633	0.9307865
R_{36}	3	5	Rotation	0.8570150	−0.5152914
D_{10}	1	3	Permutation, delay	–	–
R_{37}	3	4	Rotation	0.5043311	0.8635104
R_{38}	3	5	Rotation	0.8353180	−0.5497671

a result, we get a cascade connection of 2-D systems and a 1-D one. However, obtained structures are clearly 3-D, which means that their input and output are 3-D functions. So, to use them in real applications we need to apply any concurrent technique or sample by sample ordering to the signals and systems.

The pipeline structures, obtained in Sects. 2.1 and 3.1, have high throughput at the expense of a latency which is not larger then $(m_1(l+1)+m_2(n+1)(m+1))\tau_R$ and

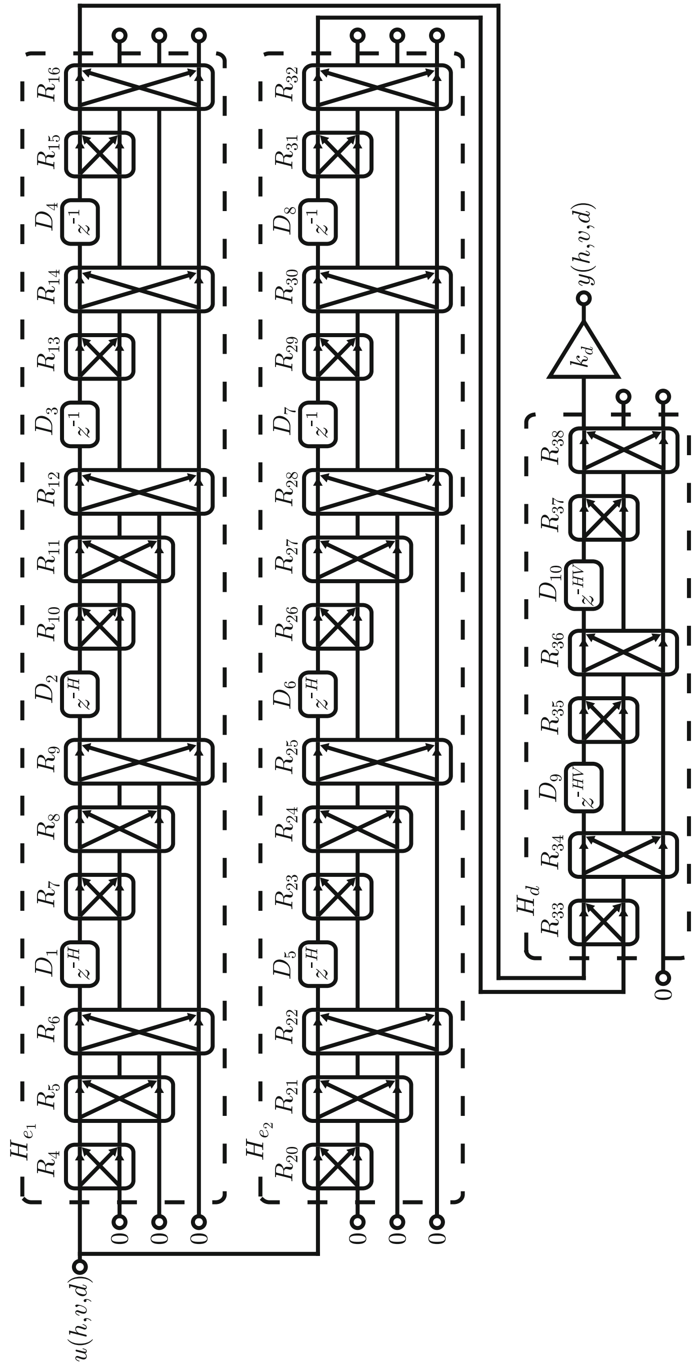


Fig. 5 Block diagram implementations of the 3-D orthogonal Laplace filter sub-blocks

$(m_1(l+1)+m_2(n+1)(m+1))\tau_R$ for separable and non-separable cases, respectively, where τ_R denotes the processing time of a Givens rotation and $m_1 \leq \min\{l+1, (n+1)\cdot(m+1)\}$, $m_2 \leq \min\{n+1, m+1\}$, for m,n, and l given by (30). So, they are well suited for hardware real-time 3-D image processing, but also may find applications in other areas where 3-D data are employed like in economy modelling, seismic data, weather forecast models, etc. It is also possible to utilise presented results as a digital replacement for analog controllers in control systems [13–15,19,20]. Due to the robust orthogonal systems properties, they will be especially applicable when high processing precision is required even for low bit quantization, as in medical imaging. One can give upper bounds for the number of Givens rotations and single delay elements. In the separable case, they are given by $n+m+l+3$ and $n+mH+lHV$, respectively, for an $H \times V$ image frames, where m, n, and l are defined in (14). For the same image in the non-separable case, a structure has no more then $m_1(l+1)+m_1m_2(n+1)(m+1)$ Givens rotations and $m_1(n+mH)+lHV$ single delay elements, for m,n, and l given by (30).

The 5 × 5 × 5 3-D Gaussian filter, presented in Sect. 2.2, has been implemented in DE2 development board with Cyclone II chip (EP2C35F672C6) running at 125 MHz clock rate [23]. In [10], a similar filter of order 3 × 3 × 3 has been presented using the same DE2 board of 100 MHz rate for RAM communication and 25 MHz for the filter module. The reported overall performance was 30 frames/s for 640 × 480 images. In [1], a 3-D anisotropic diffusion filter has been proposed which is a 5 × 5 × 5 3-D Gaussian filter. It was implemented in Stratix II chip (EP2S180F1508C4) using 7524 ALUTs and 20 DSP multipliers. Their design achieves voxel processing rate of 192–194 MHz for the system clocked at 200 MHz. An obtained precision was measured with a mean-squared error which is 0.030 for 8-bit fixed number representation.

The authors' filter, presented in [23], occupies 12020 LUTs (no DSP blocks are utilized) and achieves maximum voxel processing rate of 125 MHz which leads to 409 frames/s for 640 × 480 images. The mean-squared error for 8-bit fixed number representation is 0.002. Comparing obtained results one can see that the proposed filter voxel processing rate is at the clock rate. This is a natural property of fully pipeline structures. The number of ALUTs occupied by the proposed filter is greater than that in [1] and probably follows from the CORDIC implementation; however, we do not utilize extra DSP blocks. The proposed filter possesses good precision, due to Givens rotations, manifested by the low mean-squared error and other good properties. For more details the reader is referred to [23].

References

1. O. Dandekar, C.R. Castro-Pareja, R. Shekhar, FPGA-based real-time 3D image preprocessing for image-guided medical interventions. J. Real-Time Image Process. **1**(4), 285–301 (2007)

2. E. Depretere, P. Dewilde, Orthogonal cascade realization of real multiport digital filters. Int. J. Circuits Theory Appl. **8**, 245–272 (1980)
3. U.B. Desai, A state-space approach to orthogonal digital filters. IEEE Trans. Circuits Syst. **38**(2), 160–169 (1991)
4. DICOM sample image sets. http://www.osirix-viewer.com/datasets/ (2017)
5. D.E. Dudgeon, R.M. Mersereau, *Multidimensional Signal Processing* (Prentice-Hall, Englewood Cliffs, 1984)
6. A. Fettweis, Digital filter structures related to classical filter networks. AEÜ **25**(2), 79–89 (1971)
7. A. Fettweis, On the scattering matrix and the scattering transfer matrix of multidimensional lossless two-ports. AEÜ **36**, 374–381 (1982)
8. G.H. Golub, C.F. Van Loan, *Matrix Computations*, 3rd edn. (The Johns Hopkins Univ. Press, Baltimore, MD, 1996)
9. T. Kaczorek, *Two-Dimensional Systems* (Springer, Berlin, 1985)
10. C. Liu, H.W. Mu, D.M Liu, The realization of real-time motion tracking algorithm based on FPGA, in *Progress in Applied Sciences, Engineering and Technology*, Advanced Materials Research, vol. 926 (Trans Tech Publications, Zurich (2014), pp. 3302–3305. doi:10.4028/www.scientific.net/AMR.926-930.3302
11. P.K. Meher, J. Valls, T.B. Juang, K. Sridharan, K. Maharatna, 50 years of CORDIC: algorithms, architectures, and applications. IEEE Trans. Circuits Syst. I. Regul. Pap. **56–I**(9), 1893–1907 (2009)
12. J. Ohser, K. Schladitz, *3D Images of Materials Structures: Processing and Analysis* (Wiley-VCH, Weinheim, 2009)
13. H. Pan, X. Jing, W. Sun, Robust finite-time tracking control for nonlinear suspension systems via disturbance compensation. Mech. Syst. Signal Process. **88**, 49–61 (2017)
14. H. Pan, W. Sun, H. Gao, X. Jing, Disturbance observer-based adaptive tracking control with actuator saturation and its application. IEEE Trans. Autom. Sci. Eng. **13**(2), 868–875 (2016). doi:10.1109/TASE.2015.2414652
15. H. Pan, W. Sun, H. Gao, J. Yu, Finite-time stabilization for vehicle active suspension systems with hard constraints. IEEE Trans. Intell. Transp. Syst. **16**(5), 2663–2672 (2015). doi:10.1109/TITS.2015.2414657
16. M.S. Piekarski, Synthesis algorithm of two-dimensional orthogonal digital system: a state space approach, in *Proceedings of the XXVIIIth URSI General Assembly, New Delhi, India* (2005). www.ursi.org/Proceedings/ProcGA05/pdf/CP4.8(01080).pdf
17. J.G. Proakis, D.G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, 3rd edn. (Prentice-Hall, Upper Saddle River, 1996)
18. R.P. Roesser, A discrete state-space model for linear image processing. IEEE Trans. Autom. Control **20**(1), 1–10 (1975)
19. W. Sun, H. Pan, H. Gao, Filter-based adaptive vibration control for active vehicle suspensions with electrohydraulic actuators. IEEE Trans. Veh. Technol. **65**(6), 4619–4626 (2016). doi:10.1109/TVT.2015.2437455
20. W. Sun, S. Tang, H. Gao, J. Zhao, Two time-scale tracking control of nonholonomic wheeled mobile robots. IEEE Trans. Control Syst. Technol. **24**(6), 2059–2069 (2016). doi:10.1109/TCST.2016.2519282
21. R. Suszynski, K. Wawryn, R. Wirski, 2D image processing for auto-guiding system, in *2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS)* (2011), pp. 1–4. doi:10.1109/MWSCAS.2011.6026368
22. The SCILAB homepage. http://www.scilab.org (2017)
23. K. Wawryn, P. Poczekajlo, R. Wirski, FPGA implementation of 3-D separable Gauss filter using pipeline rotation structures, in *2015 22nd International Conference on Mixed Design of Integrated Circuits Systems (MIXDES)* (2015), pp. 589–594 . doi:10.1109/MIXDES.2015.7208592
24. R. Wirski, K. Wawryn, State space synthesis of two-dimensional FIR lossless filters, in *International Conference on Signals and Electronic Systems, 2008. ICSES '08* (2008), pp. 367–370. doi:10.1109/ICSES.2008.4673438
25. R. Wirski, K. Wawryn, Stanowa synteza systemów bezstratnych o skończonej odpowiedzi impulsowej [State-space synthesis of finite impulse response loss-less systems]. Przeglad Elektrotechniczny **86**(11a), 218–221 (2010). In Polish

26. R. Wirski, K. Wawryn, B. Strzeszewski, State-space approach to implementation of FIR systems using pipeline rotation structures, in *International Conference on Signals and Electronic Systems, 2012. ICSES '12* (2012), pp. 1–5. doi:10.1109/ICSES.2012.6382223
27. R.T. Wirski, On the realization of 2-D orthogonal state-space systems. Signal Process. **88**, 2747–2753 (2008). doi:10.1016/j.sigpro.2008.05.018
28. R.T. Wirski, Synthesis of 2-D state-space equations for orthogonal separable denominator systems, in: *International Conference on Signals and Electronic Systems (ICSES'10). Gliwice—Poland* (2010)
29. R.T. Wirski, Synthesis of orthogonal Roesser model for two-dimensional FIR filters, in *International Symposium on Information Theory and its Applications (ISITA2010), Taichung—Taiwan* (2010). doi:10.1109/ISITA.2010.5649701
30. D.C. Youla, The synthesis of networks containing lumped and distributed elements, in *Proceedings of Symposium on Generalized Networks* (Polytechnic Institute of Brooklyn Press, New York, 1966), pp. 289–343
31. L.A. Zadeh, C.A. Desoer, *Linear System Theory: The State Space Approach* (McGraw-Hill, New York, 1963)