INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2021, VOL. 67, NO. 2, PP. 295-300
Manuscript received September 14, 2020; revised April, 2021 DOI: 10.24425/ijet.2021.135979

An Overview of the Methods of Synthesis, Realization and Implementation of Orthogonal 3-D Rotation Filters and Possibilities of Further Research and Development

Paweł Poczekajło

***Abstract*—In the paper, an overview of the methods and algorithms of synthesis, realization and implementation used by the author to obtain orthogonal 3-D filters with a structure made of Givens rotations has been presented. The main advantage of orthogonal filters, which may have a lower sensitivity to quantization of the coefficients, was indicated. The author proposed a number of possible changes and modifications of individual stages, which may result in obtaining filters with even better parameters. The work will be the basis for the direction of further research.**

***Keywords*—3-D filters, orthogonal filter, rotation filter, Givens rotation, synthesis, realization, implementation, algorithms**

I. Introduction

THE publication presents and describes issues related to the synthesis, realization and implementation of orthogonal lossless 3-D filters based on Givens rotations [1]. Typically a Givens rotator is described as a system with two inputs and outputs that performs the following operation:

$$\mathbf{Y} = \mathbf{G}\mathbf{X} \tag{1}$$

where:

$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ – input data of rotator,

$\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$ – output data of rotator,

$\mathbf{G} = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix}$,

α – angle of the rotation.

On figure 1 symbol of rotator used in the article is presented.

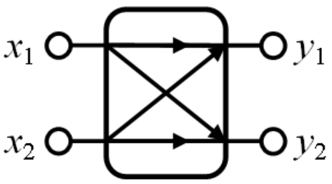

Fig. 1. Used symbol of Givens rotator.

Author is with Faculty of Electronics and Computer Science, Koszalin University of Technology, Koszalin, Poland. (e-mail: pawel.poczekajlo@tu.koszalin.pl).

It has been shown that filters in the orthogonal implementation may have better parameters than their standard implementations using convolutions [2,3]. The use of Givens rotations in such systems has been proposed, among others, in publications [3,4] and [5]. Also, the synthesis and realization of 1-D rotation systems is presented in article [6], while 2-D in [7-9] and [10]. An implementation utilizing the CORDIC algorithm, among others, is presented in [11,12] and [13]. In papers [14-16], possible implementations of orthogonal 3-D FIR filters have been described.

This article presents an overview of the methods of synthesis, realization and implementation of orthogonal 3-D rotation systems originally described by the transfer function of three variables:

$$\mathbf{H}(z_1, z_2, z_3) = \sum_{l_3=0}^{L_3} \sum_{l_2=0}^{L_2} \sum_{l_1=0}^{L_1} b_{l_1,l_2,l_3} z_1^{-l_1} z_2^{-l_2} z_3^{-l_3} \tag{2}$$

where:

L_1, L_2, L_3 – filter orders,

b_{l_1,l_2,l_3} – 3-D filter coefficients.

The following sections also propose possible directions for further research and development of appropriate methods.

II. Previous Works And Research

The work on orthogonal 3-D filters so far was based on the use of methods and functions developed for 1-D [6] and 2-D [7] systems. It is worth noting that the appropriate actions for 2-D structures are also founded on basic functions and algorithms for 1-D systems. Obtaining the final rotation structure has been divided into three stages, respectively:

- the synthesis, which includes the appropriate arithmetic operations on the original transfer function of the filter in order to obtain a paraunitary system,
- the realization, which includes the transformation of a paraunitary system to an orthogonal structure described by state-space equations with two inputs and two outputs, and obtaining of the rotation matrices,
- the implementation, which includes an appropriate interpretation of the rotation matrices and the hardware

implementation method of the system based on Givens rotations.

The author's research on the obtainment of 3-D rotation systems was presented in a consistent series of publications [14-16] and [17-19]. The methods presented there were based on the utilization of the properties of the transfer function with three variables. In line with this, a division has been made into separable and non-separable 3-D systems.

For separable 3-D FIR filters, the transfer function (2) can be written [14,16] as:

$$H(z_1,z_2,z_3)=H_1(z_1)H_2(z_2)H_3(z_3) \tag{3}$$

where:

$H_1(z_1)$, $H_2(z_2)$, $H_3(z_3)$ – 1-D sub-transfer functions of the 3-D system.

For the 3-D separable transfer function (3), the system can be realized as a cascade of 1-D sub-transfer functions (Fig. 2).

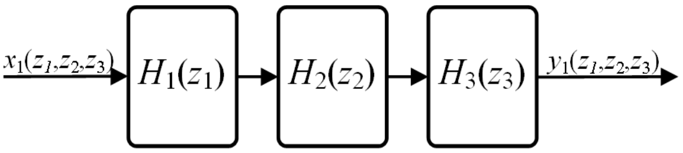

Fig. 2. 3-D system realized by a cascade of 1-D systems.

In the case of non-separable 3-D FIR filters the transfer function (2) can be written as [15]:

$$H(z_1,z_2,z_3)=\mathbf{H}_1(z_1)\mathbf{H}_{23}(z_2,z_3)=$$
$$=\left[H_{11}(z_1)\ H_{12}(z_1)\ \ldots\right]\begin{bmatrix}H_{231}(z_2,z_3)\\ H_{232}(z_2,z_3)\\ \vdots\end{bmatrix} \tag{4}$$

where:

$H_{11}(z_1)$, $H_{12}(z_1)$, … – 1-D sub-transfer functions of the 3-D system.

$H_{231}(z_2,z_3)$, $H_{232}(z_2,z_3)$, … – 2-D sub-transfer functions of the 3-D system.

For the 3-D non-separable transfer function (4), the system can be realized as a cascade of 2-D and 1-D sub-systems (Fig. 3).

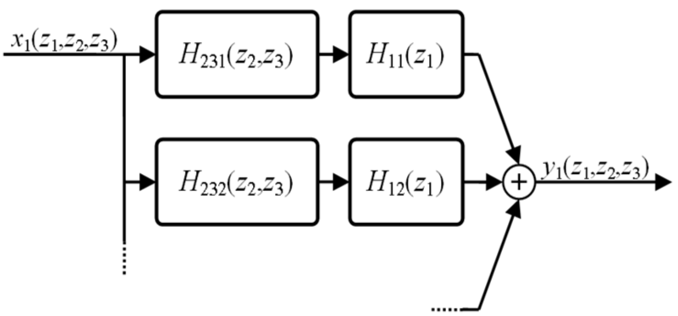

Fig. 3. 3-D system realized by a cascade of 1-D and 2-D systems.

In both (3) and (4), the original 3-D system was broken down into simpler systems of one or two variables. Therefore, further work could be carried out independently on the respective component subsystems. The 1-D and 2-D methods and functions used so far in each stage are presented in the following sections.

A. The synthesis

First of all, it is necessary to obtain a paraunitary system which must satisfy the dependency [6]:

$$\mathbf{G}^{\mathrm{T}}(z^{-1})\mathbf{G}(z)=1 \tag{5}$$

where:

$\mathbf{G}(z)=\begin{bmatrix}H(z)\\ T(z)\end{bmatrix}$,

$T(z^{-1})T(z)=1-H(z^{-1})H(z)$.

The original filter is in this case described by the transfer function $H(z)$, while the synthesis comes down to determining $T(z)$ using the factorization of polynomials [6]. In the case of two-dimensional transfer functions (4), [7] proposes an approach in which one-dimensional sub-transfer functions are obtained first:

$$H_{23\bullet}(z_2,z_3)=\left[H_{2\bullet1}(z_2)\ H_{2\bullet2}(z_2)\ \ldots\right]\begin{bmatrix}H_{3\bullet1}(z_3)\\ H_{3\bullet2}(z_3)\\ \vdots\end{bmatrix} \tag{6}$$

In this way, all one-dimensional transfer functions can be transformed into a paraunitary system independently. In the case of 3-D filters, the synthesis always comes down to finally operating on a single transfer function or a 1-D transfer function vector constituting the component subsystems of the original filter (similarly in the case of 2-D filters).

The vector of transmittance $\mathbf{H}_1(z_1)$ (4) can be realized like a one multi-input and multi-ouput system [15].

B. The realization

The paraunitary 1-D system obtained from the synthesis is in the next step transformed into orthogonal state-space equations [6]:

$$\begin{bmatrix}x(n+1)\\ y(n)\end{bmatrix}=\mathbf{S}\begin{bmatrix}x(n)\\ u(n)\end{bmatrix} \tag{7}$$

where:

$\mathbf{S}^{\mathrm{T}}\mathbf{S}=\mathbf{I}$.

State-space equations are supplemented with additional inputs, so that the matrix **S** is a square matrix [7].

In the case of a 2-D system broken down into 1-D subsystems, after determining orthogonal 1-D state-space equations, a two-dimensional Roesser model is used [20]:

$$\begin{bmatrix}x_1(n+1,m)\\ x_2(n,m+1)\\ y(n,m)\end{bmatrix}=\mathbf{S}\begin{bmatrix}x_1(n,m)\\ x_2(n,m)\\ u(n,m)\end{bmatrix}=\begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{C} & \mathbf{D}\end{bmatrix}\begin{bmatrix}x_1(n,m)\\ x_2(n,m)\\ u(n,m)\end{bmatrix} \tag{8}$$

where:

$\mathbf{A}=\begin{bmatrix}\mathbf{A}_1 & \mathbf{B}_1\mathbf{C}_2\\ 0 & \mathbf{A}_2\end{bmatrix}$,

$\mathbf{B}=\begin{bmatrix}\mathbf{B}_1\mathbf{D}_2\\ \mathbf{B}_2\end{bmatrix}$,

$\mathbf{C}=\left[\mathbf{C}_1\ \mathbf{D}_1\mathbf{C}_2\right]$,

$\mathbf{D} = [\mathbf{D}_1 \mathbf{D}_2]$,
$\mathbf{A}_1$, $\mathbf{B}_1$, $\mathbf{C}_1$, $\mathbf{D}_1$ – state-space matrices describing the first 1-D system,
$\mathbf{A}_2$, $\mathbf{B}_2$, $\mathbf{C}_2$, $\mathbf{D}_2$ – state-space matrices describing the second 1-D system.
As a result, subsystems for (3) and (4) are obtained, realized by the means of orthogonal state-space equations described by a square matrix.

Then, for a given component subsystem described by the matrix **S** (6a, 7), we decompose this matrix into the product of the rotation matrices with permutations [11,12]:

$$\mathbf{S} = \mathbf{R}_1 \mathbf{R}_2 \mathbf{R}_3 \cdots \mathbf{R}_k \tag{9}$$

where:

$$\mathbf{R}_i = \begin{array}{c} \\ \\ \\ i \\ \\ \\ k \\ k+1 \end{array} \begin{array}{c} \begin{array}{cccc} i & & k & k+1 \end{array} \\ \left[\begin{array}{ccccc} 1 & & & & \\ & 1 & & 0 & \\ & & \ddots & & \\ & 0 & & 1 & 0 \\ & & & 1 & \\ & 0 & & & \ddots \\ & \cos(\alpha_e) & & 0 & \sin(\alpha_e) \\ & -\sin(\alpha_e) & & 0 & \cos(\alpha_e) \end{array}\right] \end{array}$$

– general form of a rotation matrix with a permutation.

C. The implementation

Each of the rotation matrices $\mathbf{R}_i$ (9) can be implemented with a Givens rotation and a delay block. On figure 4 and 5 3-D systems (e.g. FIR filters) realized by Givens rotations are presented. The Givens rotation design can be tailored to the specific hardware structure. Typically, it is an FPGA that allows the rotation structure to work in a pipeline, where individual operations are performed in parallel on subsequent input data. Adopting a pipelined implementation ensures the highest possible frequency of sampling the input data, processing it and returning the output values.

The direct implementation of the Givens rotation requires multiplication and addition [13], which can be done using logic gate structures. Due to the propagation time, such a solution may result in longer data processing times by a single rotator. This will directly translate into a lower clock frequency of the finished system and thus slower input sampling and output return.

An alternative is the possibility of using DSP blocks available in selected processors (e.g. Intel Cyclone, Arria, Stratix), which can perform, among others, multiplication operations in one clock cycle and allow a higher clock frequency [15]. The condition is a sufficiently large number of available DSP blocks, because the rotation structure for a separable 3-D filter with the number of coefficients a^3 will require $12a$ multiplications and $6a$ additions in each cycle of the system operation [16] (e.g. for a filter with 5^3 coefficients, it is necessary to implement 60 multiplication blocks and 30 adder blocks). In the case of non-separable 3-D filters, the number of rotations may be greater than the number of coefficients [15], so the number of multiplications and additions, respectively, will be even greater. This is quite a significant limitation which may in some situations disqualify the possibility of using DSP blocks in the case of the selected FPGA chip.

In [11], it was proposed to use the iterative CORDIC pipeline algorithm for the implementation of Givens rotations, which eliminates the need to use multiplier blocks. This solution allows for a fully pipeline structure, with relatively simple base operations (including addition and basic logical operations). This allows the use of higher system clock frequencies (comparable to the system clock frequencies on DSP blocks [19]). However, such a solution is burdened with a greater occupancy of rotations in the FPGA processor [19]. One j-th iteration of the Givens rotation with use CORDIC algorithm is expressed as follows [11]:

$$
\begin{aligned}
q_{j+1} &= \frac{q_j}{2} \\
s_{j+1} &= \begin{cases} s_j + q_j & \text{for } s_j < 0 \\ s_j - q_j & \text{for } s_j > 0 \end{cases} \\
c_{j+1} &= \begin{cases} c_j + q_j & \text{for } c_j < 0 \\ c_j - q_j & \text{for } c_j > 0 \end{cases} \\
y_{1\,j+1} &= \begin{cases} y_{1j} + e_j & \text{for } c_j < 0,\ s_j > 0 \\ y_{1j} - e_j & \text{for } c_j > 0,\ s_j < 0 \\ y_{1j} - f_j & \text{for } c_j < 0,\ s_j < 0 \\ y_{1j} + f_j & \text{for } c_j > 0,\ s_j > 0 \end{cases} \\
y_{2\,j+1} &= \begin{cases} y_{2j} + f_j & \text{for } c_j > 0,\ s_j < 0 \\ y_{2j} - f_j & \text{for } c_j < 0,\ s_j > 0 \\ y_{2j} - e_j & \text{for } c_j < 0,\ s_j < 0 \\ y_{2j} + e_j & \text{for } c_j > 0,\ s_j > 0 \end{cases} \\
e_{j+1} &= \frac{e_j}{2} \\
f_{j+1} &= \frac{f_j}{2}
\end{aligned} \tag{10}
$$

where:
$j = 0, 1, 2, \ldots, J$,
J – number of iterations,
$q_0 = 1$,
$y_{10} = 0$,
$y_{20} = 0$,
$e_0 = x_1 - x_2$,
$f_0 = x_1 + x_2$,
$s_0 = \sin(\alpha)$,
$c_0 = \cos(\alpha)$,
$y_1 = y_{1J}$ – input data of the rotator,
$y_2 = y_{2J}$ – input data of the rotator,
α – angle of the rotation,
x_1, x_2 – input data of the rotator,
y_1, y_2 – output data of the rotator.

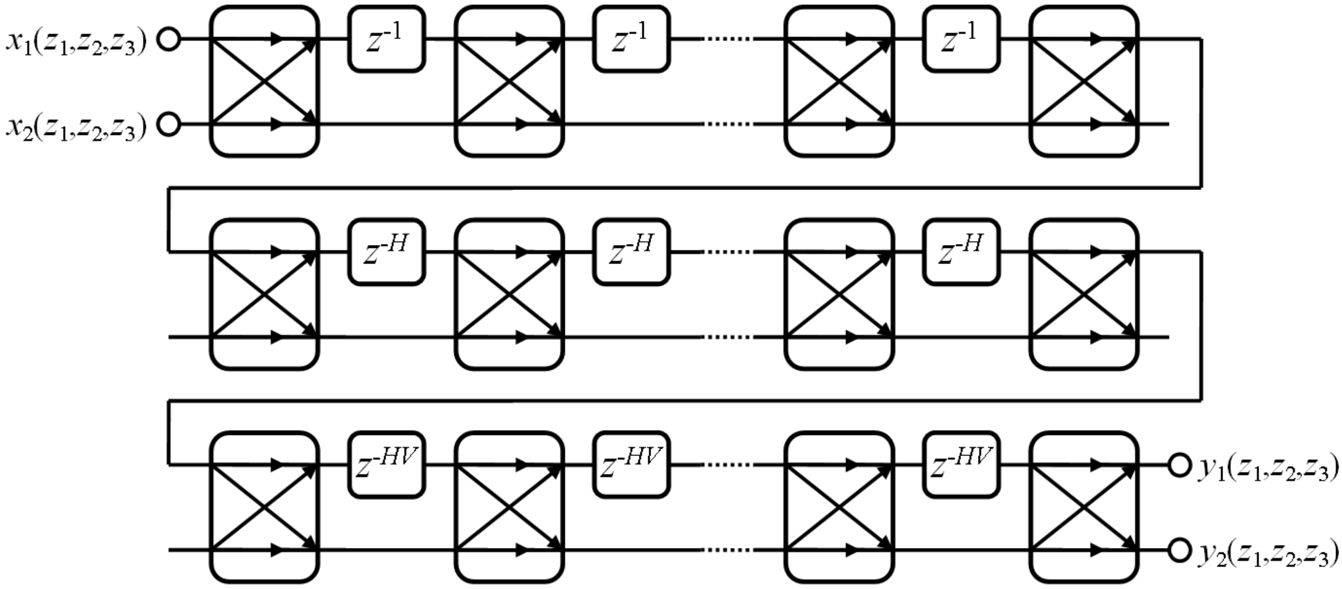

Fig. 4. 3-D separable system realized by a structure based of Givens rotators, $H \times V$ is the dimension of a single 2-D slide of a 3-D data block which is processing.

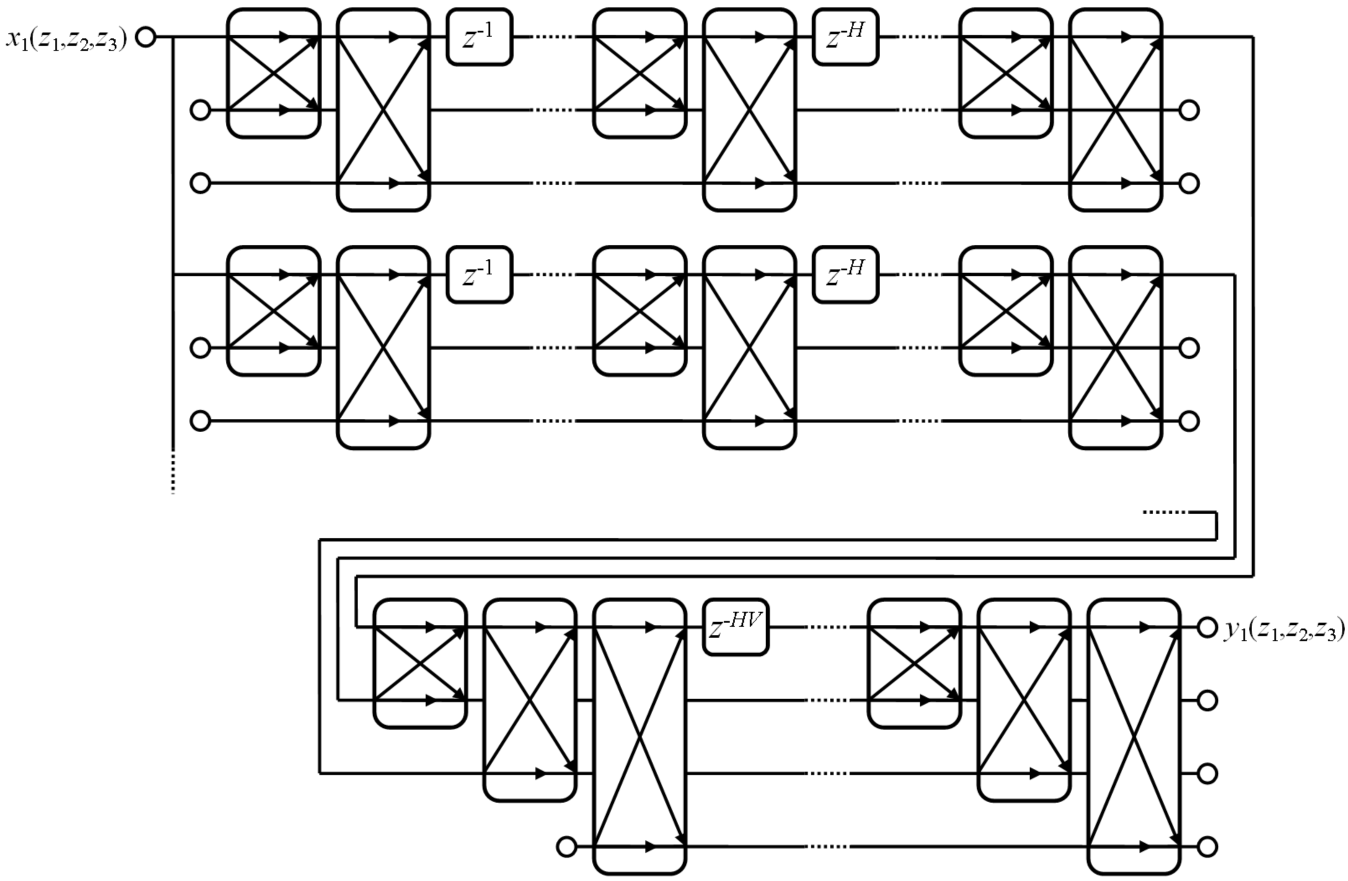

Fig. 5. 3-D separable system realized by a structure based of Givens rotators, $H \times V$ is the dimension of a single 2-D slide of a 3-D data block which is processing.

III. Possibilities For Further Development

The algorithms and solutions presented in Section 2 have been discussed in a series of consistent publications [14-19], where a number of advantages have been demonstrated, including mainly the low sensitivity of rotation filters to the quantization of coefficients [17,18]. For this reason, research into structures based on Givens rotations continues. In the following subsections, a number of proposals have been collected that constitute the possibilities of further development of appropriate algorithms and functions in order to improve their functioning and application. However, it should be emphasized that this is only a set of solutions, the verification of which, also in the context of feasibility, will be analyzed in the future.

A. The synthesis

The synthesis is a key stage and, from the point of view of numerical calculations, also quite problematic. This is the only stage where it is necessary to perform operations on polynomials. The factorization of first order polynomials is still relatively simple (even for many coefficients – with the use of appropriate computing environments).

The first significant problem, however, is the accuracy of factorization, the possible error of which may be multiplied by subsequent operations. The finally obtained structure may even have completely different parameters (profiles) than the originally installed filter. The second issue is the polynomials of many variables (in this case two or three), the factorization of which is already a significant problem. This was one of the main reasons for the decomposition of the 3-D transfer function into 1-D component subsystems according to (3), (5) and (6). The possibilities of further development of the synthesis method based on factorization may therefore come down to the development and use of more accurate algorithms and calculations as well as methods allowing for direct factorization of 2-D and 3-D polynomials (also with sufficiently high accuracy).

B. The realization

The realization so far allows to obtain orthogonal state-space equations for 1-D systems and, using the two-dimensional Roesser model, also for 2-D systems [21]. Attempts were made to obtain orthogonal state-space equations for 3-D systems in an analogous manner. Examples of such realization are presented in [22]. For non-separable 3-D systems (4), a similar implementation is also possible. However, preliminary analyzes showed that the obtained matrices were very large. The matrix A of orthogonal state-space equations for a non-separable 3-D system with several dozen coefficients, according to predictions, could have up to several thousand elements. Therefore, in the original research, this concept was abandoned as unnecessarily elaborate and problematic.

Perhaps this idea was rejected too hastily, because one of the stages of implementation is the minimization of the matrix **A** (8) [6,7], which ultimately results in a less complex structure, i.e. the more zero elements in the matrix **A**, the less rotations will be in the finally obtained system. This fact can be used at the stage of constructing orthogonal state-space matrices for non-separable 3-D systems and arranging them in such a way as to minimize it immediately. Thanks to this, despite the large number of elements in the matrices, operating on them should be quite simple and give satisfactory results.

C. The implementation

The implementation presented in Section 2.3 can be based on two methods. The first is the CORDIC algorithm, and the second is the direct execution of rotation's actions (1) using logic gates or DSP blocks. The development of work on this element may come down to changes in the CORDIC algorithm, which was presented in [18] and in a slightly modified version in [17]. The analyzes have shown that there may be situations in which the final (exact) result was obtained at the i-th iteration, and another attempt to approximate for (i+1)-th iteration may lead to a distortion of the result with an error resulting from the value of i and the assumed fixed point notation. It has also been noticed that there are register overflows (which you need to watch out for). The introduction of appropriate modifications to the CORDIC algorithm is possible, however, they significantly affect the occupancy of the structure in the FPGA and may cause an extension of the processing time. Ultimately, this can lead to a situation where the direct structure of the rotation (on the multiplication and addition blocks) is parametrically better.

Another suggestion is a completely different approach to the rotation. The concept described above is based on the fact that the rotation operates on a two-element input vector that represents the position of a given point in the Cartesian coordinate system. The rotation itself is then characterized by the values of the trigonometric functions sin(α) and cos(α), where α is the rotation angle of the rotator. An alternative may be the polar approach, in which the rotation's input is a point that is described by the length of the vector (relative to the origin of the coordinate system) and its angle. Accordingly, the rotation would also need to be described with the quantities defining the change in angle and length of the vector. This solution can provide a significant improvement in accuracy while reducing the computational complexity of the rotation.

IV. Conclusion

The article presents the methods and algorithms used to obtain rotation structures realizing 3-D filters, where the author reviewed his publications on this subject. Section 2 lists the relevant activities used so far at the stages of synthesis, realization and implementation in the hardware. Links to relevant methods for 1-D and 2-D systems were considered. Section 3 concerned the proposed changes and further development of various operations (for all stages of work) that may have a positive impact on the final structure, including by reducing calculation errors. Particular attention was paid to numerical methods and matrix operations, the ostensibly problematic performance of which may positively affect the final results. The final implementation is also an important issue, as it may affect the correct operation of the system. The article will be the basis for defining and directing the author's further scientific and research work.

References

[1] P.P. Vaidyanathan, „Multirate Systems And Filter Banks", Prentice-Hall, Englewood Cliffs, New Jersey, 1993. ISBN: 81-7758-942-3

[2] V. C. Liu, P. P. Vaidyanathan, „ Roundoff noise generated by orthogonal building blocks in signal processing structures", *IEEE International Symposium on Circuits and Systems*, Finland, 1988, pp. 2731-2734. DOI: 10.1109/ISCAS.1988.15504

[3] E. Deprettere, „Synthesis and fixed-point implementation of pipelined true orthogonal filters", *ICASSP '83. IEEE International Conference on*

Acoustics, Speech, and Signal Processing, 1983, pp. 217-220. DOI: 10.1109/ICASSP.1983.1172177

[4] A. Fettweis, „Digital filter structures related to classical filter networks", *AEÜ*, vol. 25, no. 2, 1971, pp. 79-89.

[5] E. Deprettere, „Synthesis and fixed-point implementation of pipelined true orthogonal filters", *ICASSP '83. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1983, pp. 217-220. DOI: 10.1109/ICASSP.1983.1172177

[6] R. Wirski, K. Wawryn, „Stanowa synteza systemów bezstratnych o skończonej odpowiedzi impulsowej", *Przegląd Elektrotechniczny*, vol. 86(11a), 2010, str. 218-221.

[7] R.T.Wirski, „Synthesis of 2-D state-space equations for orthogonal separable denominator systems", *International Conference on Signals and Electronic Systems (ICSES)*, Gliwice 2010, pp. 285-288.

[8] R.T.Wirski, „On the realization of 2-D orthogonal state-space systems", *Signal Processing*, vol. 88, no. 11, 2008, pp. 2747-2753. DOI:10.1016/j.sigpro.2008.05.018

[9] R. Wirski, K. Wawryn, „State space synthesis of two-dimensional FIR lossless filters", *International Conference on Signals and Electronic Systems (ICSES)*, Krakow 2008, pp. 367-370. DOI:10.1109/ICSES.2008.4673438

[10] R.T. Wirski, „Synthesis of orthogonal Roesser model for two-dimensional FIR filters", *International Symposium on Information Theory and its Applications (ISITA)*, Taichung Taiwan 2010. DOI:10.1109/ISITA.2010.5649701

[11] R.Wirski, K.Wawryn, B. Strzeszewski, „State-space approach to implementation of FIR systems using pipeline rotation structures", *International Conference on Signals and Electronic Systems (ICSES)*, Wroclaw 2012. DOI:10.1109/ICSES.2012.6382223

[12] K.Wawryn, R.Wirski, B. Strzeszewski, „Implementation of finite impulse response systems using rotation structures", *International Symposium on Information Theory and Its Applications (ISITA)*, Taichung Taiwan 2010, pp. 606-610. DOI: 10.1109/ISITA.2010.5649712

[13] R. T. Wirski, B. Strzeszewski, K. Wawryn, „Orthogonal implementation of two-dimensional separable denominator systems", *International Conference on Signals and Electronic Circuits (ICSES)*, Gliwice 2010, pp. 371-374.

[14] P. Poczekajło, R. Wirski, „Synteza separowalnych trójwymiarowych filtrów ortogonalnych o strukturze potokowej", *Przegląd Elektrotechniczny*, vol. 89(10), 2013, str. 150-152. DOI: 10.15199/48.2016.09.02

[15] P. Poczekajło, R. Wirski, „Synthesis and Realization of 3-D Orthogonal FIR Filters Using Pipeline Structures", *Circuits Systems and Signal Processing*, vol. 37, no. 4, 2018 (online 2017), pp. 1669-1691. DOI: 10.1007/s00034-017-06

[16] P. Poczekajło, K. Wawryn, „Algorithm for Realisation, Parameter Analysis, and Measurement of Pipelined Separable 3D Finite Impulse Response Filters Composed of Givens Rotation Structures", *IET Signal Processing*, vol. 12, iss. 7, 2018, pp. 857-867. DOI: 10.1049/iet spr.2017.0450

[17] K. Wawryn, P. Poczekajlo, R. Wirski, „FPGA implementation of 3-D separable Gauss filter using pipeline rotation structures", *22nd International Conference on Mixed Design of Integrated Circuits Systems (MIXDES)*, Torun 2015, pp. 589-594. DOI: 10.1109/MIXDES.2015.7208592

[18] P. Poczekajlo, K. Wawryn, „Hardware implementation of 3D pipelined laplace filter based on rotation structures", *24th International Conference on Mixed Design of Integrated Circuits Systems (MIXDES)*, Bydgoszcz 2017, pp. 276-280. DOI: 10.23919/MIXDES.2017.8005215

[19] P. Poczekajło, „Analiza wybranych metod realizacji sprzętowej rotatorów Givensa w układzie FPGA", *Przegląd Elektrotechniczny*, vol. 94(9), 2018, str. 26-28. DOI: 10.15199/48.2018.09.06

[20] R. P. Roesser, „A discrete state-space model for linear image processing", *IEEE Trans. Automat. Contr.*, vol. 20, no. 1, Feb. 1975, pp. 1–10.

[21] R. Suszyński, K. Wawryn, R. Wirski, „2D signal processing for identification and tracking moving object", *Przegląd Elektrotechniczny*, vol. 87(10), 2011, str. 126-129. ISSN 0033-2097

[22] K. Gałkowski, „State-space realisations of linear 2-D systems with extensions to the general nD (n>2) case", Springer, London, 2001. ISBN: 978-1-84628-573-8. DOI: https://doi.org/10.1007/BFb0110347