

Article

# Optimization of the FIR Filter Structure in Finite Residue Field Algebra

Dmitry Kaplun <sup>1,\*</sup>, Denis Butusov <sup>2</sup>, Valerii Ostrovskii <sup>3</sup>, Alexander Veligosha <sup>4</sup> and Vyacheslav Gulvanskii <sup>1</sup>

<sup>1</sup> Department of Automation and Control Processes, St. Petersburg Electrotechnical University “LETI”, Saint Petersburg 197376, Russia; vvgulvanskii@etu.ru

<sup>2</sup> Youth Research Institute, Saint-Petersburg Electrotechnical University “LETI”, Saint Petersburg 197376, Russia; dnbutusov@etu.ru

<sup>3</sup> Department of Computer-Aided Design, Saint-Petersburg Electrotechnical University “LETI”, Saint Petersburg 197376, Russia; vyostrovskii@etu.ru

<sup>4</sup> Department 52, Military Academy of Strategic Rocket Forces, Serpuhov 142210, Russia; aveligosha@mail.ru

\* Correspondence: dikaplun@etu.ru; Tel.: +7-921-6550470

Received: 8 October 2018; Accepted: 27 November 2018; Published: 2 December 2018



**Abstract:** This paper introduces a method for optimizing non-recursive filtering algorithms. A mathematical model of a non-recursive digital filter is proposed and a performance estimation is given. A method for optimizing the structural implementation of the modular digital filter is described. The essence of the optimization is that by using the property of the residue ring and the properties of the symmetric impulse response of the filter, it is possible to obtain a filter having almost a half the length of the impulse response compared to the traditional modular filter. A difference equation is given by calculating the output sample of modules  $p_1 \dots p_n$  in the modified modular digital filter. The performance of the modular filters was compared with the performance of positional non-recursive filters implemented on a digital signal processor. An example of the estimation of the hardware costs is shown to be required for implementing a modular digital filter with a modified structure. This paper substantiates the expediency of applying the natural redundancy of finite field algebra codes on the example of the possibility to reduce hardware costs by a factor of two. It is demonstrated that the accuracy of data processing in the modular digital filter is higher than the accuracy achieved with the implementation of filters on digital processors. The accuracy advantage of the proposed approach is shown experimentally by the construction of the frequency response of the non-recursive low-pass filters.

**Keywords:** FIR filter; modular digital filter; algebra of the finite residue field; residue number system; hardware costs; table arithmetic unit; bit capacity

## 1. Introduction

The implementation of digital signal processing (DSP) methods is effective in many practical applications, e.g., noise-resistive data processing system with high precision and resolution, easy subsystems connection, the stabilization of data processing channel etc. [1–3].

The development of DSP applications triggered many studies in a field of creating specialized DSP devices with maximal efficiency. To solve this problem, the parallel pipeline computing was introduced. This approach allowed the performing of real-time DSP tasks.

Most of the DSP tasks consider large amount of computations over the big data arrays. The increased speed of DSP calculations can be achieved through the introduction of new mathematics, based on non-positional modular codes. This paper shows the efficiency of executing DSP methods and algorithms based on ring and field algebraic structures, e.g., Residue Number System (RNS).

Common drawbacks of real-time data processing techniques caused the expansion of application areas of modular arithmetic. The analysis gives the following groups of applications for non-positional modular codes.

The first group includes traditional DSP approaches based on orthogonal signal transforms in a complex numbers field [4–9]. Parallel data processing through the computational channels, determined by RNS basis, and the low bit depth of residues allows to increase the processing speed. One paper [10] presented the algorithm for navigational data post-processing, allowing a decrease of positioning errors through the multiple measurement of pseudo-range in the chosen constellation of satellites. To post-process the signals in real-time the Katkov et al. proposed parallel computations in RNS. The implementation of the RNS approach allowed an increase in the number of measurements, which gave the desired reduction in positioning errors.

The second group includes the methods and algorithms of digital filtering based on residue number codes. Multiple studies [11–13] have described the methods for constructing non-positional digital filters that are based on algebraic structures, possessing the properties of ring and field.

In [7,13,14], Omondi et al. have shown that the digital filtering is one of the most promising applications of the finite residue field algebra, which allows overcoming the main drawback of digital filters (DF), namely, the inability to process high-frequency signals whose bandwidth is limited by the Nyquist frequency [15]. Therefore, for the high-frequency signals the analog filters were used or decimation of the processed signal was carried out. Thus, the problems of filter efficiency estimation have not been studied as intensively and the criteria for the applicability of filters based on the algebra of finite residue fields have not been developed. In addition, the structures of modular filters have been not described in detail, and there are no experimental data about their performance in various signal processing applications. From a practical point of view, the creation of DF based on the algebra of finite residue fields that allow processing of the high frequency signals will reduce application of power consumptive and hard-to-design analog filters.

Additional signal transformation, which is necessary for processing input samples of the DF during the sampling period, leads to additional distortions and a decrease of performance.

The calculation of one sample of the DF at the sampling frequency of 1 MHz is 6.08 μs for DSP processor TMS320C54. In this case, the sampling interval is approximately six times shorter than the time for calculating the output sample. When a sampling frequency is higher than the specified one, the sampling interval decreases and, consequently, the input sample cannot be processed in real time.

Let us show the comparative evaluation of the time on a certain DSP platform that is required for calculating the output sample of the positional DF and the filter implemented in the algebra of the finite residue field. The results of the evaluation are presented in Table 1.

**Table 1.** Comparative evaluation of the calculation time of the output sample of the non-recursive position filter implemented at the digital signal processing DSP and the modular digital filter at  $t_{mod2} = 2$  ns.

Type of Computation	Calculating the Output Sample Time, $t_{mod2} = 2$ ns					
	Filter Order, $N$					
	$N = 16$	$N = 32$	$N = 48$	$N = 64$	$N = 128$	$N = 256$
Modular Digital Filter	52.8	195.6	158.4	211.2	422.4	844.8
MSC8101 (Fixed Point)	26.7	53.44	80.16	108.9	217.8	435.6
ADSP TS201S (Floating Point)	2	2	2	2	2	2

The comparative evaluation was carried out for the algorithm of the non-recursive filter implemented in the algebra of the finite field under the modular (tabular) execution of its basic operation. The basic operation time was selected as 2 ns (the time of reading data from the fast-acting memory device was 0.1–4 ns [16]), while the output sample was calculated in parallel pipeline mode, i.e., in one clock cycle of the processor.

The evaluation of time taken to calculate the output sample of the DF shows the advantages of the algebra of the finite residue field over the positional numeral system.

In Section 2, implementation of modular digital non-recursive filters are introduced. In Section 3, an example of the estimation of the accuracy of calculating the output sample of the modular DF, which processes 16-bit input data and has 16-bit coefficients, has shown the correctness of the operation of the digital filtering algorithm implemented in the algebra of the finite residue field. In Section 4, a brief discussion is given. In Section 5, conclusions are made and the obtained results of calculations and simulation are presented.

## 2. Materials and Methods

The calculation of the output sample in a separate unit of the modular digital filter reads in the form [12]

$$|y_i(nT)|_{p_i} = \left| \left| |b_{i-1}|_{p_i} |x(nT - kT)|_{p_i} \right|_{p_i} + \left| |b_i|_{p_i} |x(nT - kT)|_{p_i} \right|_{p_i} \right|_{p_i}, \tag{1}$$

According to Equation (1), the coefficients  $b_i$  and the input samples  $x(nT - kT)$  of the filter must be represented by the numbers of the residue class. In this case, both the coefficients and the input samples of the filter are values belonging to the set of reminder values for the selected modulus

$$|b_i|_{p_i} \in \{0, 1, 2, \dots, \}, \tag{2}$$

$$|x_i(nT)|_{p_i} \in \{0, 1, 2, \dots, p_i - 1\}, \tag{3}$$

in accordance with Equations (2) and (3), the value of the intermediate sample of the filter will also belong to the following range of values:

$$|y_i(nT)|_{p_i} = \left| |b_{i-1}x(nT - kT)|_{p_i} + |b_i x(nT - kT)|_{p_i} \right|_{p_i} \in \{0, \dots, p_i - 1\}.$$

The sequence of arithmetic operations required to calculate the output sample of the filter, in accordance with the difference equation [17], is a chain of basic operations over the low-bit quantities (remainders)  $\alpha_i$ , described by Equation (4)

$$A = \left| \sum_{i=1}^l |F_i(\alpha_i)|_p \right|_p, \tag{4}$$

where  $F_i(\alpha_i)$  is the integer residue function of  $\alpha_i$  modulo  $p$ ;  $3 \leq l \leq n$ .

The choice of the form of the base system of the finite residue ring algebra was determined by the nature of the problem being solved and was given in [9,12–14]. For the considered problem it was important to not simply justify the method of the base system selection, but also to determine the optimal structure for the developed DF.

The mathematical model for calculating intermediate samples in the  $i$ -th unit of the modular DF in a given range can be written as:

$$\begin{aligned} |y_i(nT)|_{p_1} &= \left| |b_{i-1}x(nT - kT)|_{p_1} + |b_i x(nT - kT)|_{p_1} \right|_{p_1} \\ |y_i(nT)|_{p_2} &= \left| |b_{i-1}x(nT - kT)|_{p_2} + |b_i x(nT - kT)|_{p_2} \right|_{p_2} \\ &\dots \\ |y_i(nT)|_{p_n} &= \left| |b_{i-1}x(nT - kT)|_{p_n} + |b_i x(nT - kT)|_{p_n} \right|_{p_n} \end{aligned}$$

Then, one can obtain a mathematical model for calculating the output sample of the  $N$ -th order DF in a given range, taking into account the organization of parallel pipeline calculations in the form

$$\begin{aligned}
 |y_i(nT)|_{p_1} &= \left| |b_{i-1}x(nT - kT)|_{p_1} + |b_i x(nT - kT)|_{p_1} \right|_{p_1}, \\
 |y_i(nT)|_{p_2} &= \left| |b_{i-1}x(nT - kT)|_{p_2} + |b_i x(nT - kT)|_{p_2} \right|_{p_2}, \\
 &\dots, \\
 |y_i(nT)|_{p_n} &= \left| |b_{i-1}x(nT - kT)|_{p_n} + |b_i x(nT - kT)|_{p_n} \right|_{p_n}
 \end{aligned}
 \tag{5}$$

where:

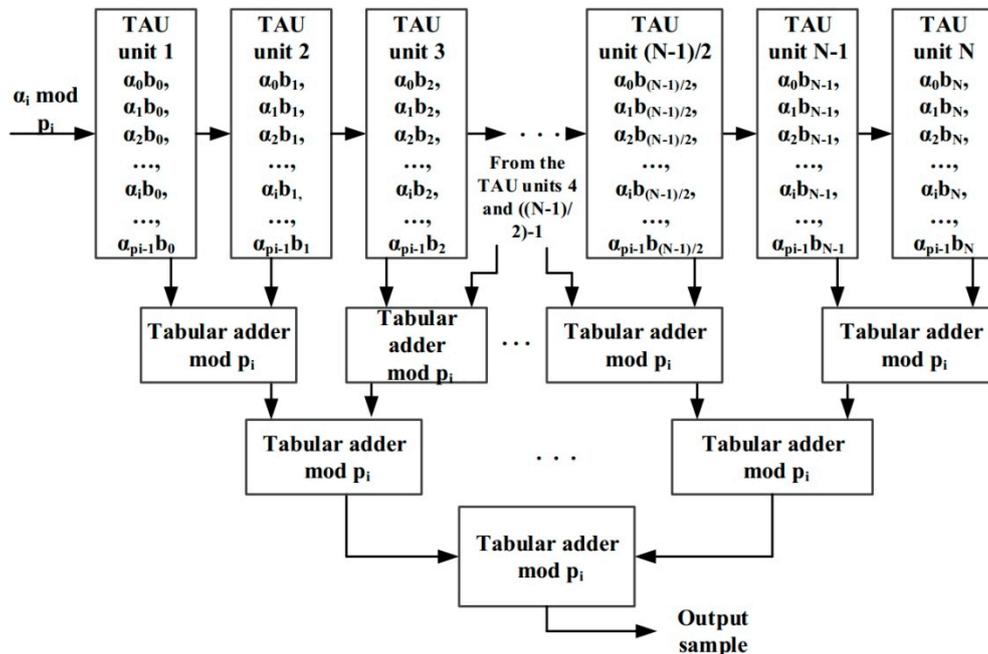
$|y^1(nT)|_{p_1} = \left| |b_0|_{p_1} |x(nT - kT)|_{p_1} + |b_1|_{p_1} |x(nT - kT)|_{p_1} \right|_{p_1}$  is the value of the intermediate sample calculated in the first two stages of the computational channel of the filter modulo  $p_1$ ;

$|y^i(nT)|_{p_1} = \left| |b_{i-1}|_{p_1} |x(nT - kT)|_{p_1} + |b_i|_{p_1} |x(nT - kT)|_{p_1} \right|_{p_1}$  is the value of the intermediate sample calculated in the  $i - 1$  and the  $i$ -th stages of the computational channel of the filter modulo  $p_1$ ;

$|y^k(nT)|_{p_1} = \left| |b_{k-2}|_{p_1} |x(nT - kT)|_{p_1} + |b_{k-1}|_{p_1} |x(nT - kT)|_{p_1} \right|_{p_1}$  is the value of the intermediate reading calculated in the penult and the last stages of the computational channel of the filter modulo  $p_1$ ; and

$|y^1(nT)|_{p_2}, |y^k(nT)|_{p_2}, |y^1(nT)|_{p_n}, |y^i(nT)|_{p_n}, |y^k(nT)|_{p_n}$  is the value of the intermediate sample calculated in the first two stages, in the  $i - 1$  and the  $i$ -th stages, in the penultimate and the last stages of the computational channel of the filter modulo  $p_2 \dots p_n$ .

The structure of the DF based on the finite residue field algebra is shown in Figure 1.



**Figure 1.** The structure of the DF operating on the basis of the finite residue field algebra. The abbreviation TAU means “Table Arithmetic Unit”.

Let us consider a non-recursive DF with a symmetric impulse response (positive symmetry). An example of a symmetric impulse response for a filter of the 13th order is shown in Figure 2. A feature of the DF with symmetric characteristic is that its samples symmetrically take values equal in magnitude and sign [18].

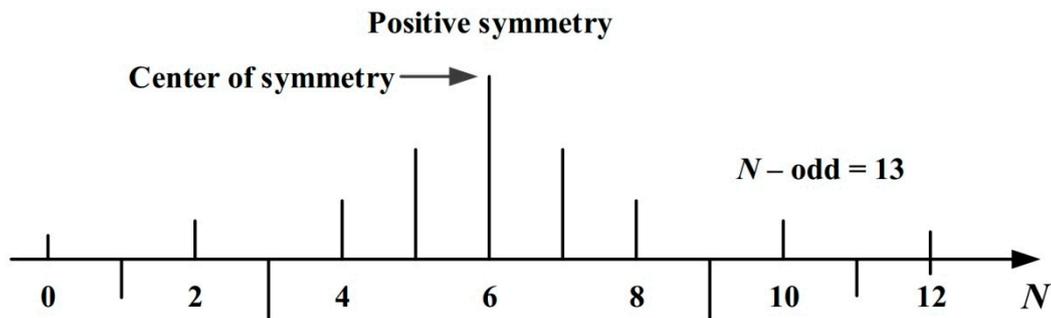


Figure 2. The symmetric impulse response of the non-recursive 13th order DF.

Therefore, the filter coefficients for the  $N$ -th order filter are defined as

$$b_0 = b_{N-1}, b_2 = b_{N-2}, \dots, b_{\frac{N-1}{2}-1} = b_{\frac{N-1}{2}+1}, \tag{6}$$

The coefficient  $b_{\frac{N-1}{2}}$  defines the center of symmetry of the impulse response of the filter and will be used only once. Using the property of cyclicity of reminders and the filter coefficients  $b_i$  modulo  $p_i$ :

$$\begin{aligned} |\alpha_i|_{p_i} &\in \{0, 1, \dots, p_i - 1\} \\ |b_i|_{p_i} &\in \{0, 1, \dots, p_i - 1\} \end{aligned} \tag{7}$$

additionally taking into account the fact that the values of the coefficients do not change during the algorithm execution, it is possible to represent the values of the intermediate samples for the coefficients  $b_0$  and  $b_{N-1}$  in the form given in Table 2.

In Table 2,  $\alpha_i$  is the value of the remainder modulo and  $p$  and  $b_i$  are the filter coefficients. One can obtain the tables similar to Table 1 for the coefficients  $b_2 = b_{N-2}, \dots, b_{\frac{N-1}{2}-1} = b_{\frac{N-1}{2}+1}$ . For the coefficient  $b_{\frac{N-1}{2}}$ , there is a separate table for performing the basic operation of a non-recursive DF. Thus, the number of tables  $K_{tmdf}$  will be determined by the value

$$K_{tmdf} = \frac{N-1}{2} + 1. \tag{8}$$

It is possible to construct a modified structure of the modular DF where the calculation of the output sample of the DF can be achieved using only  $\frac{N-1}{2} + 1$  units, which is a major advantage over the filter shown in Figure 1. The modified structure requires 1.94 times less hardware costs to implement the computing system modulo  $p_i$ , while the parameters of the filter quality do not change. For example, for a filter with a pulse characteristic length  $N_1 = 64$ , its modified structure will have a pulse characteristic length of  $N_M = 33$ ,  $N_1/N_M = 1.94$ . The structure of the modified DF based on the algebra of the finite residue field is shown in Figure 3.

Equation (6) can be re-written in the form:

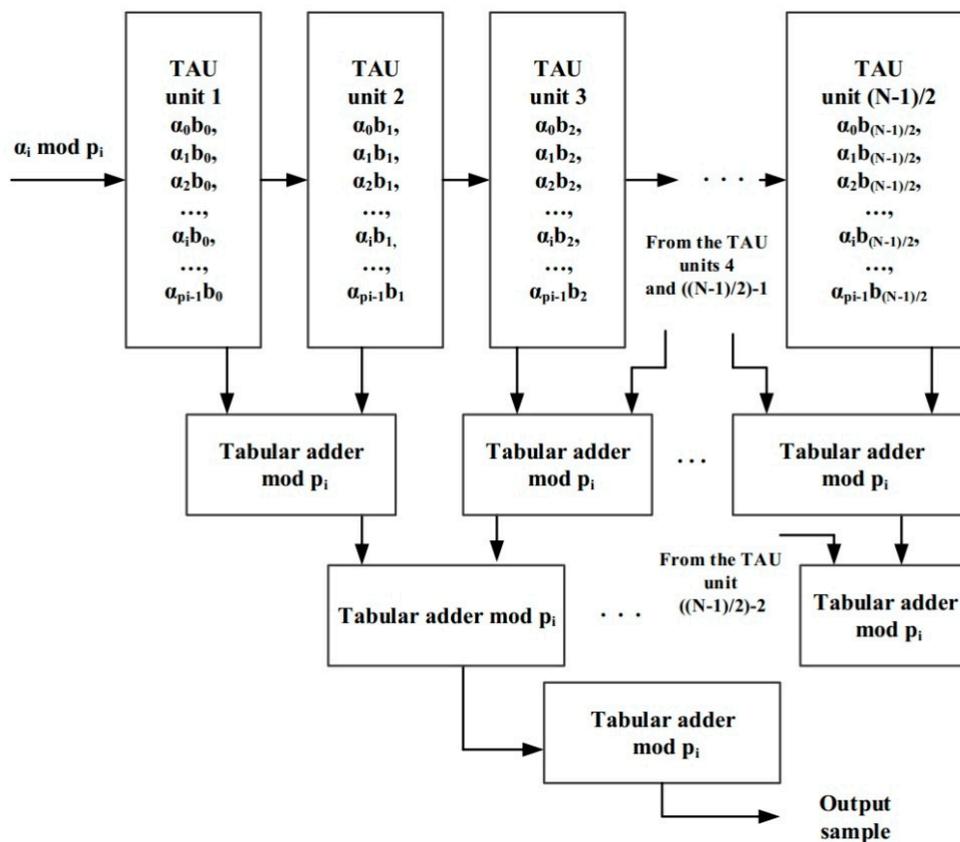
$$\begin{aligned} y|(nT)|_{p_1} &= \left| |y^1(nT)|_{p_1} + \dots + |y^i(nT)|_{p_1} + \dots + |y^k(nT)|_{p_1} \right|_{p_1} \\ y|(nT)|_{p_2} &= \left| |y^1(nT)|_{p_2} + \dots + |y^i(nT)|_{p_2} + \dots + |y^k(nT)|_{p_2} \right|_{p_2} , \\ &\dots \\ y|(nT)|_{p_n} &= \left| |y^1(nT)|_{p_n} + \dots + |y^i(nT)|_{p_n} + \dots + |y^k(nT)|_{p_n} \right|_{p_n} \end{aligned} \tag{9}$$

where:

$$\begin{aligned}
 |y^k(nT)|_{p_1} &= \left| \left| b_{\frac{N-1}{2}-1}|_{p_1} |x(nT - kT)|_{p_1} + \left| b_{\frac{N-1}{2}+1}|_{p_1} |x(nT - kT)|_{p_1} \right|_{p_1} \right|_{p_1} ; \\
 |y^k(nT)|_{p_2} &= \left| \left| b_{\frac{N-1}{2}-1}|_{p_2} |x(nT - kT)|_{p_1} + \left| b_{\frac{N-1}{2}+1}|_{p_2} |x(nT - kT)|_{p_1} \right|_{p_2} \right|_{p_2} ; \\
 &\dots \\
 |y^k(nT)|_{p_n} &= \left| \left| b_{\frac{N-1}{2}-1}|_{p_n} |x(nT - kT)|_{p_1} + \left| b_{\frac{N-1}{2}+1}|_{p_n} |x(nT - kT)|_{p_1} \right|_{p_n} \right|_{p_n} .
 \end{aligned}$$

**Table 2.** Values of the intermediate samples of the modular DF for the coefficients  $b_0$  and  $b_{N-1}$ .

$\alpha_0 b_0 + \alpha_0 b_{N-1}$	$\alpha_0 b_0 + \alpha_1 b_{N-1}$	$\alpha_0 b_0 + \alpha_2 b_{N-1}$	...	$\alpha_0 b_0 + \alpha_{p-1} b_{N-1}$
$\alpha_1 b_0 + \alpha_0 b_{N-1}$	$\alpha_1 b_0 + \alpha_1 b_{N-1}$	$\alpha_1 b_0 + \alpha_2 b_{N-1}$	...	$\alpha_1 b_0 + \alpha_{p-1} b_{N-1}$
$\alpha_2 b_0 + \alpha_0 b_{N-1}$	$\alpha_2 b_0 + \alpha_1 b_{N-1}$	$\alpha_2 b_0 + \alpha_2 b_{N-1}$	...	$\alpha_2 b_0 + \alpha_{p-1} b_{N-1}$
...	...	...	...	...
$\alpha_{p-1} b_0 + \alpha_0 b_{N-1}$	$\alpha_{p-1} b_0 + \alpha_1 b_{N-1}$	$\alpha_{p-1} b_0 + \alpha_2 b_{N-1}$	...	$\alpha_{p-1} b_0 + \alpha_{p-1} b_{N-1}$



**Figure 3.** Modified structure of the DF on the basis of the algebra of the finite residue field. DE here is a delay element, TM is a tabular multiplier, and TA is a tabular adder.

Table 3 represents an estimation of the hardware costs required to implement the DF structures shown in Figures 1 and 3, depending on the order of DF. Generally, data on the DF input comes from an analog-to-digital converter and have a bit width of 8–16 bits depending on the speed and accuracy requirements of calculations. Thus, in the traditional implementation based on the DF structures shown in Figures 1 and 3, the standard digital elements will be utilized, having a bit width of 8–16 bits.

**Table 3.** Estimation of the elements number in the structures of modular and modified modular DF.

Type of a Digital Filter		Number of Elements of a Structural Scheme of a Filter					
		Filter Order, $N$					
		$N = 16$	$N = 32$	$N = 48$	$N = 64$	$N = 128$	$N = 256$
Modular	ROM coef.	16	32	48	64	128	256
	Delay element	15	31	47	63	127	255
	Tabular multipliers	16	32	48	64	128	256
	Tabular adder	15	31	47	63	127	255
Modified modular	ROM coef.	9	17	25	33	65	127
	Delay element	9	17	24	33	65	127
	Tabular multipliers	9	17	25	33	65	127
	Tabular adder	9	17	25	33	65	127

Let us consider the processing of the 16-bit filter input samples. The data range will be from 0 to 65,535, and we can select a base system of the form:

$$p_1 = 9, p_2 = 17, p_3 = 19, p_4 = 23,$$

where the range of the representation in the finite residue field is equal to  $D_{FF} = 66,861$ . To represent the remainders for the selected bases, five digits are required ( $p_1 = 9; p_2 = 17; p_3 = 19, p_4 = 23$ ).

The required amount of memory for storing the values of the intermediate samples (Table 2) and for tree tabular adder of the 32nd order modular modified DF (Table 3) can be calculated as shown in Table 4.

**Table 4.** The required amount of memory for storing the values of the intermediate samples.

Module	Memory Capacity, Kb
$p_1$	10.77
$p_2$	47.98
$p_3$	59.94
$p_4$	7.83

The experimental evaluation was carried out as follows. As an example, we took the elements of the modified modular DF of order 32 from Table 3. For its implementation in a direct form, the typical implementation form of a FIR filter, one needs 17 tabular adders and 17 tabular multipliers. A tabular adder is a memory device that records the results of a modular addition operation of two residues for a particular modulus. Thus, for the modulus  $p_1 = 9$ , an 81-cell memory is required, where a 4-bit remainder is written in each cell. For 17 tabular adders, the total amount of required memory is 5508 bits =  $81 \times 4 \times 17$ .

In fact, the tabular multiplier represents a unit of the modular DF, and according to Table 2, the ROM coefficients and amount of memory needed for delay elements are taken into account as the hardware costs of the tabular multipliers.

The analysis has shown that in order to provide the calculation of the output sample for one modular clock cycle of the modular computational channel (the mode of parallel pipeline calculations), it is most expedient to use the adder of a tree structure.

The investigations carried out made it possible to reveal the important feature of the algebra of the finite field, namely, its natural redundancy. Since currently there is no universal element base for computational structures operating on the basis of finite field algebra, computations were performed on typical digital elements (registers, high-speed storage devices etc.).

With the application of typical digital devices, their bit capacity is equal to 16 bits for a chosen example. Therefore, the bit grid of typical digital elements was used on 33.3%, in accordance with Equation (7). Thus, the natural redundancy was inferred from the fact that for data processing it is

sufficient to use only a part of the digital grid of the device, and the remaining, large enough part of it can be utilized to optimize the structure or to perform some other procedures of finite field algebra.

Figures 1 and 3 show that the modular DF was constructed on the basis of tabular adders and tabular arithmetic blocks that represent memory devices. In order to calculate the output sample of the DF on four bases (see the example above), four computational channels were required.

When calculating the intermediate samples in the filter units, it is possible to use one digital element for two bases simultaneously. For example, if a register is used, then two values of the reminders on two bases are written in its grid, and when they are transferred to a tabular arithmetic device, the bits of the reminders can be read in parallel. Thus, when we use typical digital elements, e.g., registers for storing intermediate 16-bit results, the modulus  $p_1 = 9$  provides processing with the reminders  $\{0, 1, 2, 3, 4, 5, 6, 7, 8\}$  represented by 3–4 binary digits. Hence, we can write two numbers represented by 3–4 binary digits in one register. Note that these numbers can be reminders of different moduli.

Therefore, the total hardware (memory) costs can be reduced twice and will be modulo  $p_1$  5,4 KB; modulo  $p_2$  24 KB; modulo  $p_3$  30 KB; and modulo  $p_4$  44 KB in our example case.

A feature of modular tabular arithmetic devices is that they are symmetric [9,14] consequently, one modular tabular arithmetic unit possessed a 100% memory reserve. Let us consider a tabular multiplier modulo 5 (Table 5).

**Table 5.** Tabular multiplier modulo 5.

Remainders Modulo	Result of the Modular Multiplication			
	1	2	3	4
1	1	2	3	4
2	2	4	1	3
3	3	1	4	2
4	4	3	2	1

As one can see from Table 5, the tabular multiplier modulo 5 is symmetric with respect to the left diagonal of the table: therefore, we used only half of it during the calculations. The second half can be utilized either as a reserve or for writing an intermediate result of calculations for another modulus, which also reduces hardware costs.

It should be noted that the width of the input filter samples determines the number and magnitude of the bases of the finite field algebra and their accuracy in processing in positional computing structures. Accordingly, the accuracy of the representation of the DF input samples and coefficients determines the accuracy of the calculation of the output samples, and the accuracy of the representation of the filter coefficients also determines its frequency response and, as a consequence, its selectivity.

### 3. Results

Let us illustrate the influence of the accuracy of the filter coefficients representation to its frequency response. As an example, we chose a 15th order DF with the impulse response given in Table 6.

**Table 6.** Impulse response of the 15th order DF.

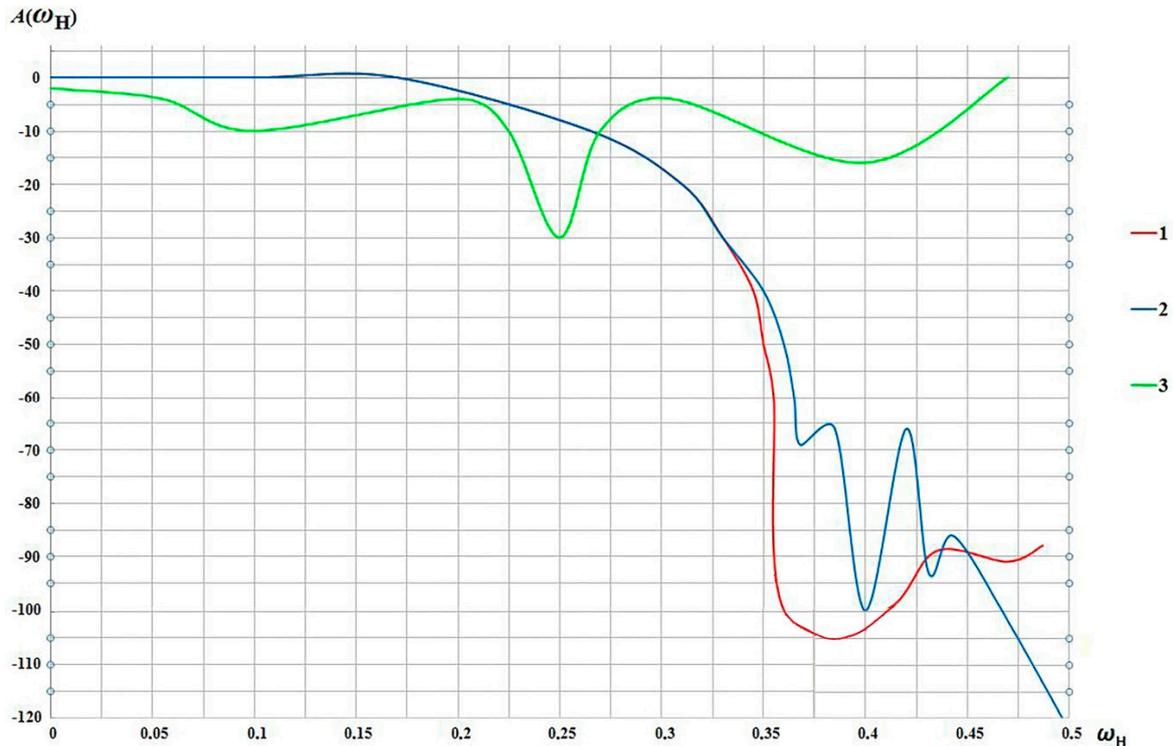
		Values of the Coefficients			
Sample numbers	1; 15	2; 14	3; 13	4; 12	
Sample values	−0.00173855	0.00329463	0.01627387	−0.01031012	
Sample numbers	5; 11	6; 10	7; 9	8	
Sample values	−0.06751172	0.01819653	0.30323396	0.47829514	

Table 7 shows the impulse response of the modular DF for the modulus  $p = 5$ .

**Table 7.** Impulse response of the 15th order DF represented in the finite field algebra.

Values of the Coefficients								
Sample numbers	1; 15	2; 14	3; 13	4; 12	5; 11	6; 10	7; 9	8
Sample values	0	3	2	2	2	3	1	4

The obtained frequency responses (FR) are shown in Figure 4.



**Figure 4.** Frequency responses of the DF: 1—frequency response (FR) of the DF with the representation of its coefficients in the algebra of the finite ring; 2—FR of the positional DF with 16-bit coefficients; 3—FR of the positional DF with 8-bit coefficients.

One can see from Figure 4 that the selectivity of the DF with the representation of its coefficients in the algebra of the finite ring overwhelms the selectivity of the positional filters. This can be explained by the fact that the filter coefficients in the algebra of the finite field can be represented with any given accuracy (see Tables 5 and 6).

Let us estimate the accuracy of calculating the output sample of a modular DF that processes 16-bit input data and has 16-bit coefficients. The range of input data will be  $D_{TCS} = 65,536$ . In order to satisfy the condition  $D_{RNS} \geq D_{TCS}$  (RNS—Residue Number System [7] for modular DF) the following system of bases  $p_1 = 2, p_2 = 5, p_3 = 7, p_4 = 11, p_5 = 13, p_6 = 19$  was chosen.

When the multiplication operation is performed in the computational channel of the modular DF, the result must be in 32-bit length, since the rounding or truncation operation cannot be applied due to the fact that the intermediate result, represented in the modular code in each computational channel, will change. Therefore, an incorrect value of the filter output will be obtained. To consider this, we need to select of an expanded system of bases:

$$p_{n+1} = 23, \quad p_{m+2} = 29, \quad p_{n+3} = 31, \quad p_{n+4} = p_{n+k} = 37.$$

Then, the extended base system takes the form:

$$p_1 = 2, \quad p_2 = 5, \quad p_3 = 7, \quad p_4 = 11, \quad p_5 = 13, \\ p_6 = 19, \quad p_7 = 23, \quad p_8 = 29, \quad p_9 = 31, \quad p_{10} = 37.$$

In this case, the structure of the DF computing device will include 10 computational channels with modulo  $p_1 \dots p_{10}$ . The modulus  $p_9 = 31$  was introduced with purpose to prevent overflow. Since the impulse response of the DF has negative coefficients, their representation in the modular code was carried out in the second half of the range  $D_{RNS}$  (numbers from 32,768 to 65,535). The input data and the coefficients of the DF, having a positive sign, were represented in the first interval in the range from 0 to 32,767. The value of the modulus  $m_1 = 2$  was chosen to simplify the operation of comparing the obtained result with the value  $D_{RNS}/2$ , i.e., determining the region of positive or negative numbers in which the result of calculations goes.

Increasing the number of bases leads to the growth of the hardware costs. In order to reduce them, we propose to scale the output samples of each computational channel of the modular DF. The scaling operation was performed by dividing the result obtained in each computational channel by the scaling factor  $D_S$ , that can be defined as

$$D_S = p_{n+1}p_{m+2}p_{n+3}p_{n+4}. \tag{10}$$

The scaling allowed us to reduce the hardware costs since the conversion of the output samples of the modular DF will be carried out not on 10 bases, but on the 6 bases  $p_1, \dots, p_6$ . The applied scaling method has been described in detail in [12].

To carry out a comparative analysis of the developed modular digital filter with the result scaling, we created its software implementations, as well as the software implementations of the positional DF for processing 16-bit data.

First, we obtained the response of the positional digital filter (Figure 5).

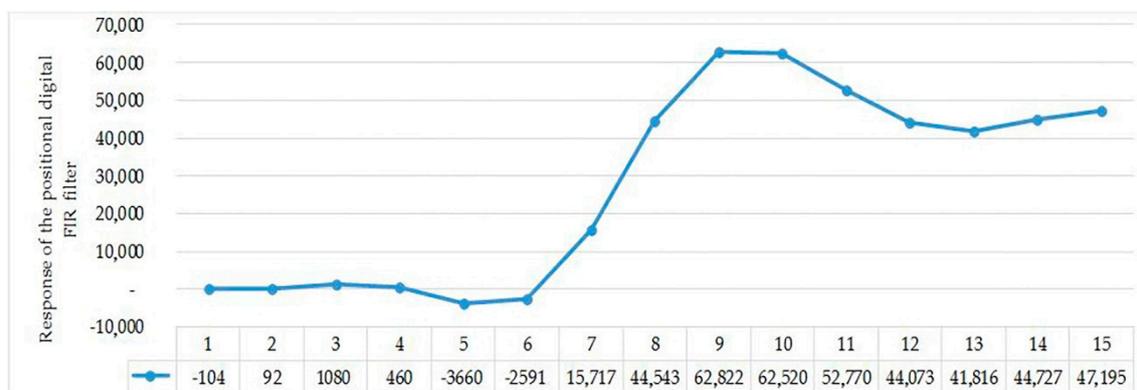


Figure 5. Response of the positional digital FIR filter.

Let us consider the situation of when the scaling method, proposed in [12], is used in the modular DF. In this case, the scaling factor is determined by the Equation (10) and for the selected RNS base system, it was  $D_S = 765,049$ .

To estimate the accuracy of calculating the output sample of the modular DF, it was necessary to multiply the obtained results by the correction factor [12]. The results of this estimation are shown in Figure 6.

The response of the DF, shown in Figure 5 (the response of the positional DF), coincides with the response shown in Figure 6 (the response of the integer modular DF using the developed scaling method). The provided example of the estimation of the accuracy of calculating the output sample of the modular DF, which processes 16-bit input data and has 16-bit coefficients, has shown the correctness of the operation of the digital filtering algorithm implemented in the algebra of the finite residue field.

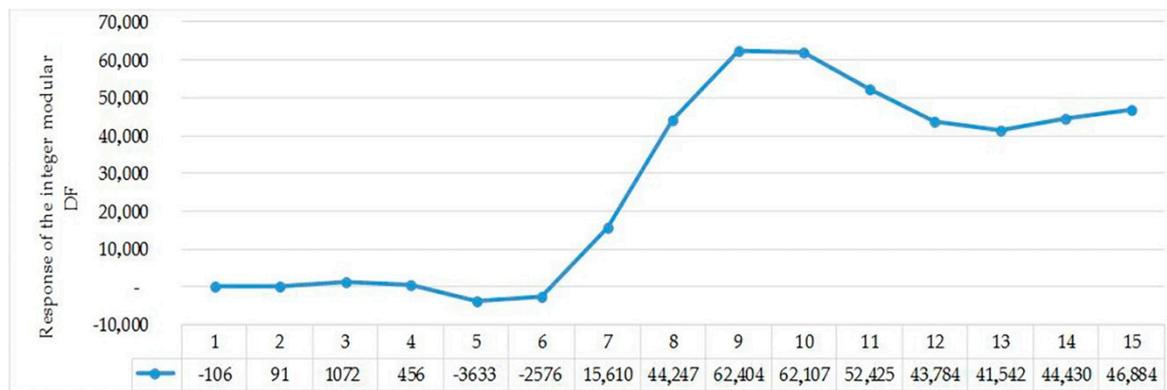


Figure 6. Response of the integer modular DF using scaling.

Another example considered in this study was the lowpass FIR filter of order 607. Its parameters are given in Table 8.

Table 8. The parameters of the test lowpass filter.

Filter order, $n$	607
Passband end, $F_{pass}$	2.7 kHz
Stopband start, $F_{stop}$	3.3 kHz
Unevenness frequency response in the passband, $A_{pass}$	1 dB
Suppression in the stopband, $A_{stop}$	120 dB
Filter squareness factor	1.22

We used multi-harmonic signal with frequencies  $f_1 = 1500$  Hz,  $f_2 = 3000$  Hz and  $f_3 = 4500$  Hz. The RNS filter has the 16 bit depth and traditional filter was implemented with double precision.

Figure 7 shows that the output signals of positional and RNS filters were almost identical. The error, calculated as a difference between outputs, was about  $10^{-14}$ . This error tends to reduce while the bit depth of RNS filters increases, therefore, it was possible to construct more efficient signal processing devices on low-precision electronics or to increase the performance of existing solutions by changing to the residue field algebra.

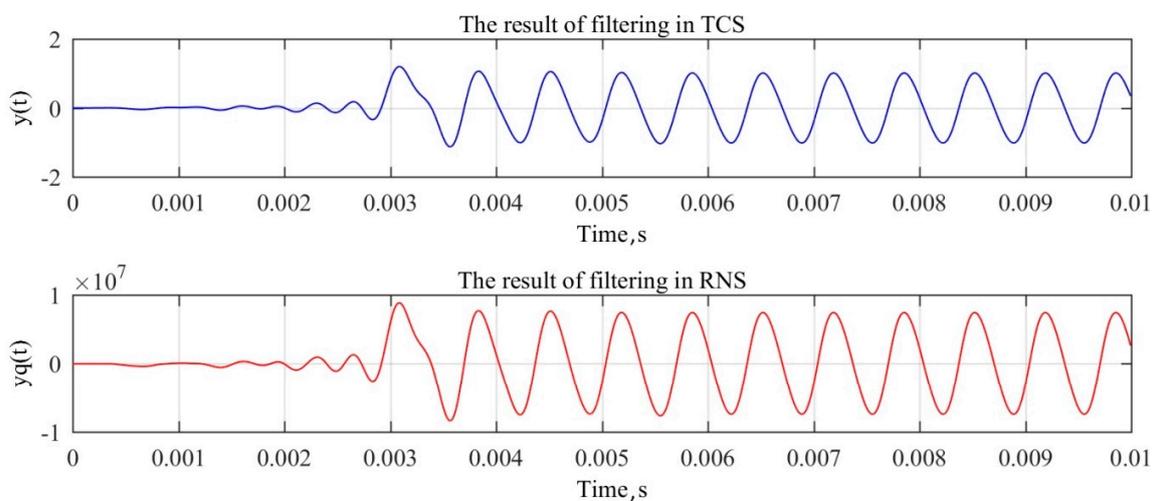
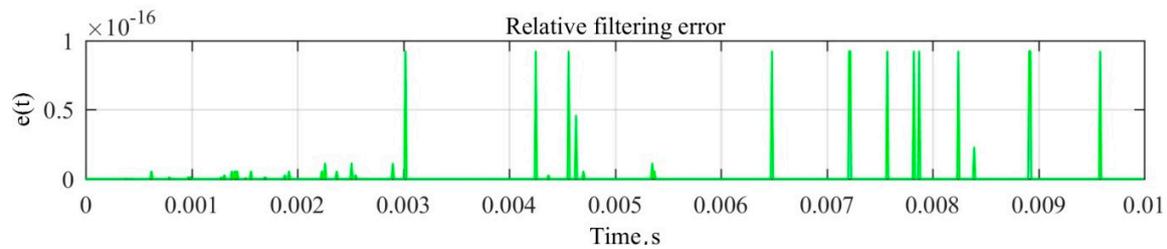


Figure 7. Cont.



**Figure 7.** The comparison of filtering error for RNS filter and positional filter.

#### 4. Discussion

Some earlier papers [19] predicted the benefits of the implementation of multistage RNS in DSP systems, however, they described no specific structures or algorithms for practical use. Our study has gone further and intently considered the tree-structured RNS and presented a multistage RNS. We described the structure of the RNS FIR filter and provided several practical tests, aimed both at a frequency characteristics study and error estimation. The optimization technique was given as well.

The obtained results open the possibility for efficient and compact hardware implementation of the digital filters on modern devices (DSP, FPGA, ASIC, etc.) for processing signals in various areas, such as radio communication, hydroacoustics, and radio- and echolocation, as well as in industry, defense, law enforcement and other fields of science and technology. Further research will be devoted to the comparison of the proposed approach with existing techniques of low-precision digital filters [20] and control systems [21] implementation based on an alternative discrete operator technique [22,23] and Gaussian approximation approach [24].

#### 5. Conclusions

The obtained results of theoretical calculations and simulation, presented in this paper, can be summarized as follows.

1. The optimization of the FIR filter structure in the algebra of finite residue field, due to the decrease in the length of its impulse response by 1.94 times, provides almost twice the reduction of the hardware costs by the number of units compared to the traditional filter structure (Figure 1).
2. In the digital filters based on the finite field algebra, the hardware costs can be significantly reduced by optimizing the filter structure and using the natural redundancy of the codes of the finite field algebra.
3. The main indicator of the quality of the filter operation is that its frequency response has higher rectangularity in comparison with positional filters and, therefore, it has better selectivity. In these terms, proposed filter structure possess better quality than existing solutions.

The obtained results show the high efficiency of the proposed FIR filter structure in residue field algebra. Results of testing the given approach in various practical applications will be reported in future papers.

**Author Contributions:** Conceptualization, D.B.; Data curation, A.V.; Formal analysis, V.O.; Investigation, D.K., D.B. and V.G.; Methodology, D.K.; Project administration, D.K.; Resources, D.B. and V.O.; Software, D.K., A.V. and V.G.; Validation, D.K., D.B., A.V. and V.G.; Visualization, A.V. and V.G.; Writing—original draft, D.K. and V.O.; Writing—review & editing, D.B. and V.O.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

#### References

1. *R&S FSQ-K96 OFDM Vector Signal Analysis with the R&S FSQ Signal Analyzer. Product Brochure, Volume 1.00;* Rohde & Schwarz: Munich, Germany, 2008.

2. Keller, T.; Liew, T.; Hanzo, L. Adaptive rate RRNS coded OFDM transmission for mobile communication channels. In Proceedings of the 51st Vehicular Technology Conference Proceedings, VTC, Tokyo, Japan, 15–18 May 2000; Volume 1, pp. 230–234.
3. Cho, Y.S.; Kim, J.; Yang, W.Y.; Kang, C.G. *MIMO-OFDM wireless communications with MATLAB*; John Wiley & Sons (Asia) Pte Ltd.: Singapore, 2010.
4. Hosseinzadeh, M.; Navi, K.; Gorgin, S. A New Moduli Set for Residue Number System:  $\{rn\ 2, rn\ 1, rn\}$ . In Proceedings of the International Conference on Electrical Engineering (ICEE), Lahore, Pakistan, 11–12 April 2007; pp. 1–6.
5. Bankas, E.K.; Gbolagade, K.A. A New Efficient FPGA Design of Residue-To-Binary Converter. *Int. J. VLSI Des. Commun. Syst. (VLSICS)* **2013**, *4*, 1–11. [[CrossRef](#)]
6. Younes, D.; Steffan, P. Universal approaches for overflow and sign detection in residue number system based on  $\{2n-1, 2n, 2n+1\}$ . In Proceedings of the Eighth International Conference on Systems (ICONS), Seville, Spain, 27 January–1 February 2013; Volume 1, pp. 77–84.
7. Omondi, A.R.; Premkumar, B. *Residue Number Systems: Theory and Implementation*; Imperial College Press: London, UK, 2007.
8. Molahosseini, A.S.; Navi, K. New Arithmetic residue to binary Converters. *Int. J. Comput. Sci. Eng. Syst.* **2007**, *1*, 295–299.
9. Cardarilli, G.C.; Nannarelli, A.; Re, M. Residue number system for low-power DSP applications. In Proceedings of the 41st Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 4–7 November 2007; pp. 1412–1416.
10. Katkov, K.A.; Kalmykov, I.A. Application of Parallel Technologies in Navigation Management under the Conditions of Artificial Ionospheric Disturbances. *World Appl. Sci. J.* **2013**, *26*, 108–113.
11. Veligosha, A.V.; Linets, G.I.; Kaplun, D.I.; Klionskiy, D.M.; Bogaevskiy, D.V. Implementation of non-positional digital filters. In Proceedings of the XIX IEEE International Conference on Soft Computing and Measurements (SCM), St. Petersburg, Russia, 25–27 May 2016; pp. 148–150. [[CrossRef](#)]
12. Kaplun, D.I.; Gulvanskiy, V.V.; Klionskiy, D.M.; Kupriyanov, M.S.; Veligosha, A.V. Implementation of digital filters in the residue number system. In Proceedings of the IEEE North West Russia Section Young Researchers in Electrical and Electronic Engineering Conference (EIconRusNW), St. Petersburg, Russia, 2–3 February 2016; pp. 220–224. [[CrossRef](#)]
13. Veligosha, A.V.; Kaplun, D.I.; Bogaevskiy, D.V.; Gulvanskiy, V.V.; Voznesenskiy, A.S.; Kalmykov, I.A. Adjustment of adaptive digital filter coefficients in modular codes. In Proceedings of the IEEE North West Russia Section Young Researchers in Electrical and Electronic Engineering Conference (EIconRusNW), Moscow, Russia, 29 January–1 February 2018; pp. 1167–1170. [[CrossRef](#)]
14. Stamenkovic, N. Digital FIR filter architecture based on the residue number system. *Facta Univ. Ser. Electron. Energ.* **2009**, *22*, 125–140. [[CrossRef](#)]
15. Antoniou, A. *Digital Filters: Analysis, Design, and Applications*; McGraw-Hill: New York, NY, USA, 1993.
16. Kalmykov, I.A.; Veligosha, A.V.; Kaplun, D.I.; Klionskiy, D.M.; Gulvanskiy, V.V. Parallel-pipeline implementation of digital signal processing techniques based on modular codes. In Proceedings of the XIX conference on Soft Computing and Measurements (SCM), St. Petersburg, Russia, 25–27 May 2016; pp. 213–214. [[CrossRef](#)]
17. Veligosha, A.V.; Kaplun, D.I.; Klionskiy, D.M.; Bogaevskiy, D.V.; Gulvanskiy, V.V.; Kalmykov, I.A. Error Correction of Digital Signal Processing Devices using Non-Positional Modular Codes. *Autom. Control Comput. Sci.* **2017**, *51*, 167–173. [[CrossRef](#)]
18. Mitra, S.K. *Digital Signal Processing: A Computer-Based Approach*; McGraw-Hill: New York, NY, USA, 1998; 940p.
19. Chervyakov, N.I.; Lyakhov, P.A.; Shulzhenko, K.S. FIR Filters in Two-Stage Residue Number System. In Proceedings of the International Conference on Engineering and Telecommunication, Moscow, Russia, 26–28 November 2014; pp. 26–29.
20. Karimov, T.I.; Butusov, D.N.; Andreev, V.S.; Rybin, V.G.; Kaplun, D.I. Compact Fixed-Point Filter Implementation. In Proceedings of the 22nd Conference of FRUCT Association, Jyväskylä, Finland, 15–18 May 2018; Volume 2, pp. 73–78.
21. Middleton, R.H.; Goodwin, G.C. Improved Finite Word Length Characteristics in Digital Control Using Delta Operators. *IEEE Trans. Autom. Control* **1986**, *31*, 1015–1021. [[CrossRef](#)]

22. Butusov, D.N.; Karimov, T.I.; Kaplun, D.I.; Karimov, A.I. Delta operator filter design for hydroacoustic tasks. In Proceedings of the 6th Mediterranean Conference on Embedded Computing (MECO), Bar, Montenegro, 1–15 June 2017; pp. 1–4.
23. Kauraniemi, J.; Laakso, T.I.; Hartimo, I. Delta Operator Realizations of Direct-Form IIR Filters. *IEEE Trans. Circuits Syst. II Analog Digit. Signal Process.* **1998**, *45*, 41–52. [[CrossRef](#)]
24. Capizzi, G.; Coco, S.; Sciuto, G.L.; Napoli, C. A New Iterative FIR Filter Design Approach Using a Gaussian Approximation. *IEEE Signal Process. Lett.* **2018**, *25*, 1615–1619. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).