applied sciences

MDPI

Article

Error-Correction Coding Using Polynomial Residue Number System

Igor Anatolyevich Kalmykov [1], Vladimir Petrovich Pashintsev [1], Kamil Talyatovich Tyncherov [2], Aleksandr Anatolyevich Olenev [3] and Nikita Konstantinovich Chistousov [1,*]

[1] Department of Information Security of Automated Systems, North-Caucasus Federal University Stavropol, 1 Pushkina Str., 355017 Stavropol, Russia; kia762@yandex.ru (I.A.K.); pashintsevp@mail.ru (V.P.P.)
[2] Information Technologies, Mathematics and Natural Sciences, Ufa State Petroleum Technological University, 1 Kosmonavtov St., 450064 Ufa, Russia; academic-mvd@mail.ru
[3] Stavropol State Pedagogical Institute, 417 Lenina Str., 355009 Stavropol, Russia; olenevalexandr@gmail.com
* Correspondence: chistousov.nik@yandex.ru

Abstract: There has been a tendency to use the theory of finite Galois fields, or GF(2n), in cryptographic ciphers (AES, Kuznyechik) and digital signal processing (DSP) systems. It is advisable to use modular codes of the polynomial residue number system (PRNS). Modular codes of PRNS are arithmetic codes in which addition, subtraction and multiplication operations are performed in parallel on the bases of the code, which are irreducible polynomials. In this case, the operands are small-bit residues. However, the independence of calculations on the bases of the code and the lack of data exchange between the residues can serve as the basis for constructing codes of PRNS capable of detecting and correcting errors that occur during calculations. The article will consider the principles of constructing redundant codes of the polynomial residue number system. The results of the study of codes of PRNS with minimal redundancy are presented. It is shown that these codes are only able to detect an error in the code combination of PRNS. It is proposed to use two control bases, the use of which allows us to correct an error in any residue of the code combination, in order to increase the error-correction abilities of the code of the polynomial residue number system. Therefore, the development of an algorithm for detecting and correcting errors in the code of the polynomial residue number system, which allows for performing this procedure based on modular operations that are effectively implemented in codes of PRNS, is an urgent task.

Keywords: residue number system (RNS); polynomial residue number system (PRNS); detecting and correcting errors; positional characteristic; polynomial interval

Citation: Kalmykov, I.A.; Pashintsev, V.P.; Tyncherov, K.T.; Olenev, A.A.; Chistousov, N.K. Error-Correction Coding with Usage of Polynomial Residue Number System. *Appl. Sci.* **2022**, *12*, 3365. https://doi.org/10.3390/app12073365

Academic Editors: Aleksandr Cariow and Oleg Finko

Received: 11 February 2022
Accepted: 24 March 2022
Published: 25 March 2022

1. Introduction

Modular codes occupy a special place among arithmetical codes that allow for the maximal possible performance of computing devices. There are two types of modular codes:

- codes of residue number system (RNS);
- codes of polynomial residue number system (PRNS).

The basic principles for constructing modular codes of residue number system were first described in several books [1–4]. It was shown in these works that codes of RNS can increase the speed with which arithmetical operations (addition, subtraction and multiplication) can be performed. This is due to the fact that multi-digit integers are replaced by the corresponding tuple of residues in codes of RNS, which are obtained by dividing these operands by n bases of the code, which are relatively prime numbers. At the same time, residues are processed in parallel and independently of each other on bases of the code. The works [5–7] show methods for performing modular addition operation as well

as their circuit solutions. Basic principles of circuit implementations of high-speed modular multipliers are presented in [8–10].

Codes of RNS have found application in digital signal processing (DSP) systems due to their high speed. The works [11–14] present mathematical models of special DSP as well as their circuit solutions, use of which allows for orthogonal signal transformations in real time.

Mathematical models of adaptive and classical digital filters implemented in the code of RNS were developed in order to improve the quality of signal processing. The works [15,16] present algorithms for performing signal filtering, in which the use of a code of RNS makes it possible to increase the speed of the calculating filter's response. The work [17] shows methods for constructing a fault-tolerant digital filter functioning in codes of RNS. Implementation of adaptive filters using codes of residue number system is presented in [18–20]. An algorithm for choosing coefficients of a digital filter using integral arithmetic of codes of residue number system was developed in [21] and provides zero error in the response of the filter.

Codes of RNS have found application in neural systems. Neural networks using a finite ring neural network model were developed using the parallel functioning principle of neural networks and combining it with parallel principles of computation in a code of RNS [22–24]. Use of this model allowed for an increase in the speed and accuracy of neural networks.

One of the areas of effective application of codes of RNS is cryptography. As a rule, the residue number system is used in cryptographic systems with a public key. The expediency of using codes of RNS in RSA encryption algorithm is proved in the works [25–29]. Use of codes of RNS aims to create a method for fast and efficient encryption based on elliptical cryptography while ensuring high cryptographic strength of the cipher and the integrity of the transmitted information.

However, modular codes of RNS are able to increase both the speed of calculations and to detect and correct errors that occur during calculations due to failures in the operation of the computing device. The theory of constructing redundant codes of RNS is considered in detail in [2–4]. These works show fundamental differences between redundant codes of RNS and error-correction codes; r additional control bases are introduced to detect and correct errors in the code of RNS, which is specified by k informational bases. As a result, the code combination is expanded by r residues. At the same time, control bases are equal in relation to informational bases. This is due to the fact that all residues contain information about the original integer. Therefore, control bases are involved in performing arithmetical operations as well as informational ones.

As a rule, control symbols are obtained in error-correction codes by summing modulo two of corresponding informational bits. In this case, there is no possibility of simultaneous execution of arithmetical operations on both the informational and control parts of the error-correction code.

Therefore, redundant arithmetical codes of RNS have found wide application in the construction of fault-tolerant DSP systems, as the usage of redundant arithmetical codes of RNS allows for the operability of the computing device to be maintained by detecting and correcting errors that occur during data processing. For example, the works [30–33] consider the issues with building a fault-tolerant special processor operating in codes of RNS. The issues with increasing the fault tolerance of digital filters functioning in codes of RNS are considered in [34–37].

Algorithms for detecting and correcting errors in the codes of the residue number system are based on positional characteristics (PC), which show the location of an integer represented as a redundant code combination of RNS relative to the range of allowed code combinations (the allowed range), which is determined by the product of k informational bases. If the number is within this range, then the code combination does not contain an error; otherwise the code combination is erroneous.

One of the features of redundant modular codes of RNS is their multivariance of positional characteristics (PC). One of these characteristics is the number interval. Algorithms for calculating this PC, as well as their circuit implementations, are provided in [2,3,38–41]. Residues are sequentially excluded from the code combination and then compared with the allowed range in the method of projections. If all reduced code combinations (obtained during transformations) are less than the allowed range (i.e., they are within this range), then the code combination of RNS is an allowed one. Otherwise, all reduced code combinations will be greater than the allowed range except for one in which the erroneous residue is removed.

The method of the extending the bases of a code of RNS used for error detection and correction is presented in [42,43]. This method is based on the equality of informational and control residues, as they are all obtained from the original integer by calculating its residues' modulo bases of the code of RNS; r control residues are calculated using k informational residues. Then, the calculated residues are compared with control residues in the code combination. If they all match, then there is no error in the code combination of RNS.

In the works [44,45], the use of higher coefficients of mixed radix conversion (MRC) is proposed as a positional characteristic. This system is used to perform reverse conversion from a code of RNS to a positional code. If the higher MRC's coefficients are zero, this means that the code combination of RNS does not contain an error.

Summarizing what has been said to this point, we can conclude that the theoretical foundations for the construction of redundant codes of RNS capable of detecting and correcting calculation errors are well developed, and that is why codes of RNS have found application in the construction of fault-tolerant high-speed DSP special processors.

Codes of polynomial residue number system received their name based on the fact that irreducible polynomials are used as bases of the modular code [46–48]. It is necessary to represent an integer as a polynomial and then calculate residues of this polynomial's modulo bases of PRNS in order to obtain a code combination of PRNS. As the code is defined in the polynomial ring, there are modular operations for it (addition, subtraction and multiplication), which are performed in parallel and independently on the bases using small-bit residues [49,50]. Such principles of constructing codes of PRNS make it possible to increase the speed of calculations. Therefore, these codes are used when orthogonal transformations of signals are being performed in real time [51]. New algorithms for constructing modular multipliers, as well as their circuit implementations, are proposed in [52–54] for reducing the execution time of multiplicative operations in PRNS.

Unlike codes of RNS, codes of polynomial residue number system have found wide application in cryptography, as shown in [55–58]. The work [59] presents a technique that allows for switching from the AES encryption algorithm implemented in the finite field GF(2^8) to calculations in the ring of finite fields of a smaller order, GF(2^4). It is known that the AES block cipher is implemented in the Galois field GF(2^8) using a generating polynomial $p(z) = z^8 + z^4 + z^3 + z + 1$. The present work proposes to perform all AES cryptographic transformations using two irreducible polynomials, $p_1(z) = z^4 + z + 1$ and $p_2(z) = z^4 + z^3 + 1$. In this case, the plaintext byte will be represented as two residues, which are obtained when it is divided by the polynomials $p_1(z) = z^4 + z + 1$ and $p_2(z) = z^4 + z^3 + 1$. As a result, the encryption process using the AES algorithm is performed using the code of polynomial residue number system, which has two informational bases. Circuit implementations of this technique are given in the works [60–63].

As failures may occur during the operation of computing devices using elements of finite Galois fields GF(2^n), it becomes necessary to use codes that are able to detect and correct these errors.

Currently, error-correction BCH codes, which belong to polynomial cyclic codes, are widely used in signal processing and transmission systems. BCH codes are capable of correcting multiple errors. The construction of such codes is based on the theory of finite Galois fields GF(2^n). Depending on the multiplicity of the error being corrected, developers determine the number of irreducible polynomials that are used to obtain the generating polynomial. For example, we can use the code (15, 11) and the generating polynomial $g(x) = x^4 + x + 1$ in order to correct a single error, which is understood as a distortion of one binary digit in a code combination. This code is defined over the Galois field GF(2^4). It is necessary to use the code (15, 7) in order to correct two errors in the code combination. The product of two minimal polynomials, $x^4 + x + 1$ and $x^4 + x^3 + 1$, will provide the generating polynomial $g(x) = x^8 + x^7 + x^5 + x^4 + x^3 + x + 1$ of the considered code.

However, there is a difference between codes of PRNS and BCH codes; the latter does not allow the use of polynomial cyclic codes for detecting and correcting errors that occur during calculations. All residues in modular codes of PRNS, both informational and redundant, are obtained by dividing the original polynomial into bases of the code, which are irreducible polynomials. Because the process of obtaining the residues is the same, we can say that they are equal in relation to each other. Therefore, codes of PRNS are arithmetical codes and can be used when calculations are being performed. At the same time, arithmetical operations will take place in the same way on all bases of the code. It is the equality of residues in relation to each other that allows us to dynamically change the number of informational and control bases. For example, the following irreducible polynomials can be defined for GF(2^4): $m_0(x) = x + 1$, $m_1(x) = x^4 + x + 1$, $m_3(x) = x^4 + x^3 + x^2 + x + 1$, $m_5(x) = x^2 + x + 1$, $m_7(x) = x^4 + x^3 + 1$. If it is necessary to detect a single error, which is understood as a distortion of one residue of the code, then one base of the code can be taken as a control base, for example, $m_7(x) = x^4 + x^3 + 1$. The four remaining bases of the code of PRNS will be considered informational. If a single error needs to be corrected during the calculation process, a polynomial, $m_1(x) = x^4 + x + 1$, can be added to the control bases. Then, the code will have three informational bases and two control bases. At the same time, in the BCH code it is not necessary to make any recalculations such as shown above with the generating polynomial.

BCH codes are not arithmetical. In BCH codes, control symbols are determined by summing modulo two corresponding information digits. In this case, informational and control bits are not equal in relation to each other. If calculations are performed in BCH codes, then the corresponding arithmetical operations can be performed only on informational bits. However, these operations cannot be performed on control bits, as the result will lead to incorrect control bits. As BCH codes are not arithmetical, they cannot be used for detection and correction of errors that occur during calculations [64,65].

However, the issues of building redundant codes of PRNS have not yet found wide application. This disadvantage hinders the expansion of the scope of application of codes of polynomial residue number system. Therefore, the objectives of the present article are:

- To conduct studies on the error-correction abilities of redundant codes of polynomial residue number system with one and two control bases.
- To develop an algorithm for detecting and correcting errors in redundant codes of PRNS based on the calculation of a positional characteristic (a polynomial interval) in which only modular operations are used, that is, without performing a reverse conversion from a code of PRNS to a positional code and then performing a division operation.

2. Modular Codes

Modular codes belong to the class of arithmetical codes. The name of these codes is associated with the rules of their construction: arithmetical operations are performed modulo the bases of the code. As mentioned above, modular codes are of two types, and we will consider both of them here.

2.1. Codes of Residue Number System

Relatively prime numbers, $m_1, m_2, \ldots, m_n$, are used as bases in a code of RNS. The integer D is represented in a code of RNS as a tuple of residues:

$$X = (x_1, x_2, \ldots, x_n) \tag{1}$$

where $x_i \equiv X \bmod m_i$ and $i = 1, 2, \ldots, n$.

The product of the bases determines the range of possible code combinations, which is called the operating range:

$$W_n = \prod_{i=1}^{n} m_i \tag{2}$$

In order to unambiguously represent the integer X as a code combination of RNS, the following condition must be met:

$$X < W_n \tag{3}$$

The Chinese Remainder Theorem gives a bijection between an integer X that satisfies Condition (3), along with a residues vector, $(x_1, x_2, \ldots, x_n)$. Thanks to this property, it is possible to perform calculations with sets of the short-length residues of long numbers rather than the long numbers themselves. At the same time, calculations for each of the modules can be performed in parallel [1–4]:

$$U \bullet X = ((u_1 \bullet x_1) \bmod m_1, \ldots, (u_n \bullet x_n) \bmod m_n) \tag{4}$$

where $\bullet$ are addition, subtraction, and multiplication operations, $U \equiv u_i \bmod m_i$, and $i = 1, 2, \ldots, n$.

Codes of RNS are able to increase the performance of arithmetic devices because calculations occur in parallel on the bases of the code and the operands have a small bit depth.

2.2. Codes of Polynomial Residue Number System

The bases are irreducible polynomials, $p_i(z)$, $i = 1, \ldots, n$, in a code of polynomial residue number system. It is necessary to convert an integer, D, presented in a binary number system into a polynomial, $D(z)$, prior to its conversion into the code combination of PRNS. After that, residues obtained by dividing the polynomial $D(z)$ by bases of the code of PRNS are found in [2–4]:

$$D(z) = (d_1(z), d_2(z), \ldots, d_n(z)) \tag{5}$$

where $d_i(z) \equiv D(z) \bmod p_i(z)$ and $i = 1, \ldots, n$.

The operating range of the code of polynomial residue number system is defined as

$$W_n(z) = \prod_{i=1}^{n} p_i(z) \tag{6}$$

Then, using the isomorphism generated by the Chinese Remainder Theorem (CRT) in polynomials, we obtain

$$D(z) \bullet U(z) = ((d_1(z) \bullet u_1(z)) \bmod p_1(z), ..., (d_n(z) \bullet u_n(z)) \bmod p_n(z)) \quad (7)$$

where ● are addition, subtraction, and multiplication operations, $U(z) \equiv u_i(z) \bmod p_i(z)$, and $i = 1, 2, ..., n$.

Despite the similar principles of implementing codes of RNS and codes of PRNS, the latter have not found as wide application due to the insufficient level of development of the theoretical foundations for constructing redundant codes of polynomial residue number system that are capable of detecting and correcting errors that occur during calculations in the polynomial ring.

3. Theoretical Foundations of Building Redundant Codes of Polynomial Residue Number System

3.1. PRNS Codes With One Control Base

The basis of the construction of error-correction codes is the introduction of redundancy, which allows us to increase the total number of all possible combinations and obtain two disjointed subsets. The first subset contains allowed code combinations, and the second contains prohibited code combinations. We will use this approach for construction of error-correction codes of polynomial residue number system.

A similar result can be achieved if the set of allowed code combinations of PRNS is reduced [2–4]. In this case, *j* bases are chosen from *n* bases of the code of PRNS, which will be considered redundant. Thus, there is a decrease in the number of allowed code combinations. Then, the new range of allowed code combinations of PRNS will be determined as

$$P_j(z) = \frac{W_n(z)}{\prod_j p_j(z)} \quad (8)$$

where $j = 1, ..., n$.

It can be concluded based on this equality that an increase in the number of redundant bases leads to a decrease in the subset of allowed code combinations for which $\deg D_1(z) < \deg P_j(z)$. The remaining code combinations of the code of PRNS will belong to prohibited code combinations for which $\deg D_2(z) \geq \deg P_j(z)$.

This can be illustrated by an example.

Example 1. Let four bases of the code of PRNS be given as $p_1(z) = z^5 + z^4 + z^3 + z^2 + 1$ $p_2(z) = z^5 + z^3 + z^2 + z + 1$, $p_3(z) = z^5 + z^2 + 1$, $p_4(z) = z^5 + z^3 + 1$. Then, the operating range of the code of polynomial residue number system is determined by Expression (6), and is equal to

$$W_4(z) = \prod_{i=1}^{4} p_i(z) = z^{20} + z^{19} + z^{18} + z^{17} + z^{15} + z^{12} + z^9 + z^8 + z^5 + z^3 + z^2 + z + 1$$

Take one base, $p_4(z) = z^5 + z^3 + 1$, and divide the operating range $W_4(z)$ by it. As a result, the original operating range, $W_4(z)$, will be divided into two subsets, namely, allowed code combinations and prohibited code combinations, and the range of allowed code combinations will be equal to

$$P_4(z) = \frac{W_4(z)}{p_4(z)} = z^{15} + z^{14} + z^9 + z^6 + z^5 + z^4 + z^2 + z + 1$$

Thus, allowed code combinations of PRNS will be those that satisfy the condition $\deg D_1(z) < \deg P_4(z)$, while the condition $\deg D_2(z) \geq \deg P_4(z)$ will be valid for prohibited code combinations of PRNS.

Take two bases, $p_4(z) = z^5 + z^3 + 1$ and $p_3(z) = z^5 + z^2 + 1$. This will reduce the subset of allowed code combinations:

$$P_{3,4}(z) = \frac{W_4(z)}{p_3(z)p_4(z)} = z^{10} + z^9 + z^7 + z^6 + z^5 + z^4 + z^3 + z + 1$$

Then, allowed code combinations of PRNS will be those that satisfy the condition $\deg D_1(z) < \deg P_{3,4}(z)$, while the condition $\deg D_2(z) \geq \deg P_{3,4}(z)$ will be valid for prohibited code combinations of PRNS.

The following conclusion can be drawn based on the above example: we can set a value $P_j(z)$ by choosing an appropriate set of j bases of the modular code to determine all allowed code combinations of PRNS, and usage of j redundant bases allows us to detect and correct errors that occur in code combinations of PRNS. At the same time, it is obvious that if the condition $\deg D_1(z) < \deg P_j(z)$ is met, then $\left[\frac{D_1(z)}{P_j(z)}\right] = 0$, and if the condition $\deg D_2(z) \geq \deg P_j(z)$ is met, then $\left[\frac{D_2(z)}{P_j(z)}\right] > 0$, where $\left[\frac{f(z)}{g(z)}\right]$ is the quotient obtained from the result of polynomial long division of polynomial $f(z)$ by polynomial $g(z)$.

Consider a special case in which only one base, $p_i(z)$, of the code of polynomial residue number system is used. Then,

$$P_i(z) = \frac{W_n(z)}{p_i(z)} \tag{9}$$

Using the last equality, we obtain a set of code combinations in which residue modulo $p_i(z)$ is being changed. These code combinations can be presented as

$$D_{iR}(z) = (d_1(z), \ldots, d_{i-1}(z), R_i(z), d_{i+1}(z), \ldots, d_n(z)) \tag{10}$$

where $R_i(z) = \left\{0, 1, z, \ldots, z^{\deg p_i(z)-1} + z^{\deg p_i(z)-2} + \ldots + z + 1\right\}$ are residues modulo $p_i(z)$.

Then, using the CRT isomorphism in polynomials, it is necessary to prove that two code combinations, $D_{iR}(z) = (d_1(z), \ldots, R_i(z), \ldots, d_n(z))$ and $D_{iV}(z) = (d_1(z), \ldots, V_i(z), \ldots, d_n(z))$, in which residues modulo the i-th base differ from each other ($V_i(z) \neq R_i(z)$), will provide different results when they are divided by $P_i(z)$, or, in other words, $\left[\frac{D_{iR}(z)}{P_i(z)}\right] - \left[\frac{D_{iV}(z)}{P_i(z)}\right] \neq 0$. Fulfillment of this property is the condition that will allow us to detect and correct errors using PRNS.

Theorem 1. If we have a code of polynomial residue number system with bases $p_1(z), p_2(z), \ldots, p_n(z)$ where $p_1(z), p_2(z), \ldots, p_n(z)$ are irreducible polynomials, then for code combinations $D_{iR}(z) = (d_1(z), \ldots, d_{i-1}(z), R_i(z), d_{i+1}(z), \ldots, d_n(z))$ and $D_{iV}(z) = (d_1(z), \ldots, d_{i-1}(z), V_i(z), d_{i+1}(z), \ldots, d_n(z))$ where $V_i(z) \neq R_i(z)$ and $\deg\{V_i(z), R_i(z)\} < \deg P_i(z)$, the following condition is met:

$$\left[\frac{D_{iR}(z)}{P_i(z)}\right]-\left[\frac{D_{iV}(z)}{P_i(z)}\right]\neq 0 \tag{11}$$

where $P_i(z)=\frac{W_n(z)}{p_i(z)}$, $W_n(z)=\prod_{i=1}^{n} p_i(z)$ is the operating range of the code of PRNS, and $i=1, 2,..., n$.

Proof of Theorem 1. According to the Chinese Remainder Theorem in polynomials, orthogonal bases are used to perform an inverse conversion from a code of PRNS to a positional code, defined as

$$B_i(z)=m_i(z)\frac{W_n(z)}{p_i(z)} \tag{12}$$

where $W_n(z)=\prod_{i=1}^{n} p_i(z)$ and $m_i(z)$ is the weight of the *i*-th orthogonal basis such that $\mathrm{m_i}(z)\frac{W_n(z)}{p_i(z)}\equiv 1 \bmod p_i(z)$. □

Then, the difference between two code combinations of PRNS $D_{iR}(z)$ and $D_{iV}(z)$ translated into a positional code is determined by

$$D_{iR}(z)-D_{iV}(z)=\left|R_i(z)-V_i(z)\right|_{p_i(z)} m_i(z)\frac{W_n(z)}{p_i(z)} \tag{13}$$

where $\left|f(z)\right|_{g(z)}$ is a notation for the expression $f(z)\bmod g(z)$.

If Condition (11) is met, then the following equality is valid:

$$\left|R_i(z)-V_i(z)\right|_{p_i(z)} m_i(z)\frac{W_n(z)}{p_i(z)}=k(z)\frac{W_n(z)}{p_i(z)} \tag{14}$$

where *k(z)* is a coefficient and $\mathrm{k}(z)\neq 0$.

By transforming (reducing) the left and right sides of the equality, we obtain

$$k(z)=\left|R_i(z)-V_i(z)\right|_{p_i(z)} m_i(z) \tag{15}$$

If the weight of the orthogonal basis is $\mathrm{m_i}(z)=1$, then Expression (15) will take the form

$$k(z)=\left|R_i(z)-V_i(z)\right|_{p_i(z)} \tag{16}$$

As $V_i(z)\neq R_i(z)$ and $\mathrm{k}(z)\neq 0$, we obtain

$$V_i(z)=\left|R_i(z)-k(z)\right|_{p_i(z)} \tag{17}$$

Using the CRT for polynomials and converting the code combinations into a positional code, we obtain

$$D_{iR}(z)=\left|d_1(z)B_1(z)+...+R_i(z)B_i(z)+...+d_n(z)B_n(z)\right|_{W_n(z)} \tag{18}$$

For the second combination,

$$\begin{aligned} D_{iV}(z) &= \left|d_1(z)B_1(z)+\ldots+V_i(z)B_i(z)+\ldots+d_n(z)B_n(z)\right|_{W_n(z)} = \\ &= \left|d_1(z)B_1(z)+\ldots+\left|R_i(z)-k(z)\right|_{p_i(z)}B_i(z)+\ldots+d_n(z)B_n(z)\right|_{W_n(z)} = \\ &= \left|d_1(z)B_1(z)+\ldots+R_i(z)B_i(z)+\ldots+d_n(z)B_n(z)\right|_{W_n(z)} - \left|k(z)B_i\right|_{W_n(z)} = \\ &= D_{iR}(z) - \left|k(z)B_i\right|_{W_n(z)} \end{aligned} \tag{19}$$

Therefore, the following inequality holds:

$$\left[\frac{D_{iR}(z)}{P_i(z)}\right] - \left[\frac{D_{iV}(z)}{P_i(z)}\right] \neq 0 \tag{20}$$

If the weight of the orthogonal basis is $m_i(z) \neq 1$, then

$$\left|R(z)-V(z)\right|^{+}_{p_i(z)} = \left|\frac{k(z)}{m_i(z)}\right|_{p_i(z)} \tag{21}$$

As $m_i(z) \neq 0$ and $k(z) \neq 0$, the right side of Equality (21) is a nonzero element $\beta_i(z)$, which is a residue modulo $p_i(z)$. Then, we obtain

$$\begin{aligned} D_{iV}(z) &= \left|d_1(z)B_1(z)+\ldots+V_i(z)B_i(z)+\ldots+d_n(z)B_n(z)\right|_{W_n(z)} = \\ &= \left|d_1(z)B_1(z)+\ldots+\left|R_i(z)-\beta_i(z)\right|_{p_i(z)}B_i(z)+\ldots+d_n(z)B_n(z)\right|_{W_n(z)} = \\ &= \left|d_1(z)B_1(z)+\ldots+R_i(z)B_i(z)+\ldots+d_n(z)B_n(z)\right|_{W_n(z)} - \left|\beta_i(z)B_i\right|_{W_n(z)} = \\ &= D_{iR}(z) - \left|\beta_i(z)B_i\right|_{W_n(z)} \end{aligned} \tag{22}$$

Therefore, the following inequality holds:

$$\left[\frac{D_{iR}(z)}{P_i(z)}\right] - \left[\frac{D_{iV}(z)}{P_i(z)}\right] \neq 0 \tag{23}$$

The theorem is proved.

Theorem 2. If an inequality

$$\left[\frac{D_{iR}(z)}{P_i(z)}\right] \neq \left[\frac{D_{iV}(z)}{P_i(z)}\right] \tag{24}$$

exists for two code combinations $D_{iR}(z)$ and $D_{iV}(z)$ presented in the code of polynomial residue number system, then the following inequality holds:

$$R_i(z) \neq V_i(z) \bmod p_i(z) \tag{25}$$

Proof of Theorem 2. The difference between two code combinations $D_{iR}(z) = (d_1(z),\ldots,R_i(z),\ldots,d_n(z))$ and $D_{iV}(z) = (d_1(z),\ldots,V_i(z),\ldots,d_n(z))$ is determined by the following expression:

$$D_{iR}(z) - D_{iV}(z) = \left| R_i(z) - V_i(z) \right|_{p_i(z)} m_i(z) \frac{W_n(z)}{p_i(z)} \quad (26)$$

Given Inequality (24), we can conclude that the first multiplier of Expression (26) is not zero. Then,

$$R_i(z) \neq V_i(z) \bmod p_i(z)$$

The theorem is proved. □

Consider the application of the theorem in the example of a code of PRNS with various bases.

Example 2. The code of PRNS is set with the following bases: $p_1(z) = z+1$, $p_2(z) = z^2 + z + 1$, $p_3(z) = z^4 + z^3 + z^2 + z + 1$, $p_4(z) = z^4 + z^3 + 1$, $p_5(z) = z^4 + z + 1$. The operating range of the code is $W_5(z) = \prod_{i=1}^{5} p_i(z) = z^{15} + 1$. The orthogonal bases for this modular code will have the following form:

$$B_1(z) = z^{14} + z^{13} + z^{12} + z^{11} + z^{10} + z^9 + z^8 + z^7 + z^6 + z^5 + z^4 + z^3 + z^2 + z + 1$$

$$B_2(z) = z^{14} + z^{13} + z^{11} + z^{10} + z^8 + z^7 + z^5 + z^4 + z^2 + z$$

$$B_3(z) = z^{14} + z^{13} + z^{12} + z^{11} + z^9 + z^8 + z^7 + z^6 + z^4 + z^3 + z^2 + z$$

$$B_4(z) = z^{14} + z^{13} + z^{12} + z^{11} + z^9 + z^7 + z^6 + z^3$$

$$B_5(z) = z^{12} + z^9 + z^8 + z^6 + z^4 + z^3 + z^2 + z$$

Consider the zero code combination $D(z) = (0, 0, 0, 0, 0) = 0$. Let $i = 2$; then, we obtain the following code combinations:

$$D_{2,1}(z) = (0, 1, 0, 0, 0) = z^{14} + z^{13} + z^{11} + z^{10} + z^8 + z^7 + z^5 + z^4 + z^2 + z$$

$$D_{2,z}(z) = (0, z, 0, 0, 0) = z^{14} + z^{12} + z^{11} + z^9 + z^8 + z^6 + z^5 + z^3 + z^2 + 1$$

$$D_{2,z+1}(z) = (0, z+1, 0, 0, 0) = z^{13} + z^{12} + z^{10} + z^9 + z^7 + z^6 + z^4 + z^3 + z + 1$$

Consider $P_2(z) = \dfrac{W_5(z)}{p_2(z)} = z^{13} + z^{12} + z^{10} + z^9 + z^7 + z^6 + z^4 + z^3 + z + 1$.

Then, we have $\left[\dfrac{D_{2,0}(z)}{P_2(z)} \right] = 0$, $\left[\dfrac{D_{2,1}(z)}{P_2(z)} \right] = z$, $\left[\dfrac{D_{2,z}(z)}{P_2(z)} \right] = z+1$, $\left[\dfrac{D_{2,z+1}(z)}{P_2(z)} \right] = 1$.

It can be concluded based on the analysis of the application of Theorems 1 and 2 for other bases of the code that if the degree of polynomials $D(z)$ presented as code combinations of PRNS does not exceed deg $P_i(z)$, then $\left[\frac{D(z)}{P_i(z)}\right]=0$. For other polynomials, those with $\deg D(z) \geq \deg P_i(z)$, we have $\left[\frac{D(z)}{P_i(z)}\right]\neq 0$.

The theorems and examples presented above demonstrate the potential possibility of constructing redundant codes of polynomial residue number system that are capable of detecting and correcting errors that distort residues in the code combination. In order to do this, it is necessary to choose bases $p_j(z)$ in such a way that a polynomial of the form

$$P_j(z)=\left[\frac{W_n(z)}{\prod_j p_j(z)}\right] \tag{27}$$

is obtained such that its degree is greater than the degree of the polynomial $D(z)$, on the basis of which allowed the code combinations of PRNS can be constructed. Then, the remaining polynomials presented in the code of polynomial residue number system that satisfy the condition $(\deg C(z) \geq \deg P_j(z)) \cap (\deg C(z) < \deg W_n(z))$ form a subset of prohibited code combinations of PRNS.

It is necessary to determine minimal number of control bases introduced into the modular code in order to solve the problem of constructing a code of PRNS capable of detecting and correcting errors that occur in the residues of the code combination of PRNS. This will allow for the development of algorithms for correcting erroneous residues in the code of polynomial residue number system, which can arise due to failures in the calculation process, while introducing a minimal amount of necessary redundancy.

Minimal redundancy of the code of polynomial residue number system is achieved by introducing one control base. For an ordered bases system used in PRNS, the redundant base $p_{n+1}(z)$ must satisfy the following condition:

$$\deg p_1(z) \leq \deg p_2(z) \leq \ldots \leq \deg p_n(z) \leq \deg p_{n+1}(z) \tag{28}$$

In this case, the range of the code is changed to the value

$$W_{n+1}(z)=\prod_{i=1}^{n+1} p_i(z)=W_n(z)p_{n+1}(z) \tag{29}$$

In this case, $W_n(z)$ is considered the operating range of the code and $W_{n+1}(z)$ is the full range. The following rule is adopted in order to detect and correct errors in the code of polynomial residue number system. Consider polynomials represented as $A(z)=(\alpha_1(z),\alpha_2(z),\ldots,\alpha_{n+1}(z))$ and $C(z)=(c_1(z),c_2(z),\ldots,c_{n+1}(z))$ for which the following conditions are met:

$$\deg A(z) < \deg W_n(z), \quad \deg C(z) < \deg W_n(z) \tag{30}$$

If the degree of the result of performing modular arithmetic operations (modular addition and multiplication) in PRNS, namely,

$$D(z)=A(z)\bullet C(z)=(d_1(z),d_2(z),\ldots d_{n+1}(z)) \tag{31}$$

does not exceed the degree of the operating range

$$\deg D(z) < \deg W_n(z) \tag{32}$$

then the original code combinations and the result do not contain errors. Otherwise, the code combination of PRNS contains errors and is prohibited.

Theorem 3. If the condition

$$\deg D^*(z) \geq \deg W_n(z) \tag{33}$$

is satisfied for a polynomial $D^*(z)$ that corresponds to a code combination of the ordered code of polynomial residue number system with bases $p_1(z),\ldots,p_n(z),p_{n+1}(z)$, then the code combination for such a polynomial in PRNS contains at least one error. In other words, $D^*(z)$ represents a prohibited code combination.

Proof of Theorem 3. Let a single error in the code combination of PRNS occur on the *i*-th base, where $i = 1, 2,\ldots, n+1$. This leads to a distortion of the residue, as follows:

$$d_i^*(z) = \left| d_i(z) + \Delta d_i(z) \right|_{p_i(z)} \tag{34}$$

where $\Delta d_i(z)$ is an error depth and $\deg(\Delta d_i(z)) < \deg p_i(z)$. □

Then, the erroneous combination of PRNS has the form

$$D^*(z) = (d_1(z),\ldots, d_i^*(z),\ldots,d_{n+1}(z)) \tag{35}$$

By way of contradiction, assume that the polynomial $D^*(z)$ presented in the code of polynomial residue number system does not contain an error. Then, we use the Chinese Remainder Theorem for polynomials and perform the inverse conversion:

$$D^*(z) = \left| d_1(z)B_1(z) + \ldots + d_i^*(z)B_i(z) + \ldots + d_{k+1}(z)B_{k+1}(z) \right|_{W_{n+1}(z)} \tag{36}$$

At the same time, according to Theorem 1, there is a polynomial $D(z) = (d_1(z),\ldots, d_i(z),\ldots, d_{n+1}(z))$ among the allowed code combinations of PRNS that differs from $D^*(z)$ by residue $d_i(z)$ on the *i*-th base. Furthermore, the condition $\deg D(z) < \deg W_n(z)$ holds for such a code combination.

Then, based on the CRT, the given polynomial is presented as

$$D(z) = \left| d_1(z)B_1(z) + \ldots + d_i(z)B_i(z) + \ldots + d_{k+1}(z)B_{k+1}(z) \right|_{W_{n+1}(z)} \tag{37}$$

Therefore, the following equality is valid:

$$\left[D^*(z) / W_n(z) \right] = \left[D(z) / W_n(z) \right] \tag{38}$$

Substitute Expressions (36) and (37) into Equality (38) and, taking into account the properties of orthogonal bases of redundant PRNS, that is,

$$B_i(z) = m_i(z)\frac{W_{n+1}(z)}{p_i(z)} = m_i(z)\frac{W_n(z)p_{n+1}(z)}{p_i(z)} \tag{39}$$

we obtain

$$\frac{d_i^*(z)m_i(z)p_{n+1}(z)}{p_i(z)} = \frac{d_i(z)m_i(z)p_{n+1}(z)}{p_i(z)} \tag{40}$$

Equality (40) is satisfied under the condition $d_i(z) = \left|d_i(z) + \Delta d_i(z)\right|_{p_i(z)}$, in other words, when an error depth is $\Delta d_i(z) = 0$.

However, according to the provided information, an error occurred in the code combination of PRNS and the residue is $d_i^*(z) \neq d_i(z)$. Therefore, the code combination $D_i^*(z)$ contains an error on the *i*-th base of the code of polynomial residue number system and Condition (33) is valid for it.

The proof is finished.

It is obvious that Theorem 3 underlies the principles of constructing redundant codes of polynomial residue number system that are capable of detecting and correcting errors in calculations. Using this theorem, it we have shown that the distortion of any residue of the code of polynomial residue number system transforms an allowed code combination into a prohibited one. It is the violation of Condition (30) that makes it possible to detect an error in the code combination of PRNS. Consider the issue of determining the location of the error in the PRNS code combination. We can use the following theorem in order to do this.

Theorem 4. If a combination $D^*(z) = (d_1(z),..., d_i^*(z),...,d_{n+1}(z))$ has a single error, $d_i^*(z) = \left|d_i(z) + \Delta d_i(z)\right|_{p_i(z)}$, on the *i*-th base in an ordered code of polynomial residue number system, then the polynomial interval corresponding to this error is determined by the expression

$$S(z) = \left[\frac{\Delta d_i(z) m_i(z) p_{k+1}(z)}{p_i(z)}\right] \tag{41}$$

Proof of Theorem 4. A single error in the redundant code of polynomial residue number system means a distortion of one residue in the code combination. Accordingly, if an error occurs on the *i*-th base of the code, an erroneous polynomial, $D^*(z) = (d_1(z),..., d_i^*(z),...,d_{n+1}(z))$, is obtained by adding the allowed code combination $D(z) = (d_1(z),..., d_i(z),..., d_{n+1}(z))$ and $d_i^*(z) = \left|d_i(z) + \Delta d_i(z)\right|_{p_i(z)}$, where $\Delta d_i(z)$ is an error depth**.** □

Use the CRT to convert an erroneous combination of PRNS into a positional code. Then, we have

$$\begin{aligned} D^*(z) &= \left|d_1(z)B_1(z) + ... + d_i^*(z)B_i(z) + ... + d_{n+1}(z)B_{n+1}(z)\right|_{W_{n+1}(z)} = \\ &= \left|d_1(z)B_1(z) + ... + \left|d_i(z) + \Delta d_i(z)\right|_{p_i(z)}^{+}(z)B_i(z) + ... + d_{n+1}(z)B_{n+1}(z)\right|_{W_{n+1}(z)} = \\ &= D(z) + \left|\Delta d_i(z)B_i(z)\right|_{W_{n+1}(z)}, \end{aligned} \tag{42}$$

where $B_i(z)$ is the *i*-th orthogonal basis of PRNS.

The polynomial interval of the code combination of PRNS is determined by the expression

$$S(z) = \left[\frac{D(z)}{W_n(z)}\right] \tag{43}$$

From the Expression (43), we can see that if $\deg D(z) < \deg W_n(z)$, then $S(z) = 0$.

Find the polynomial interval for the erroneous combination:

$$S(z)=\left[\frac{D^*(z)}{W_n(z)}\right]=\left[\frac{D(z)+\left|\Delta d_i(z)B_i(z)\right|^+_{W_{n+1}(z)}}{W_n(z)}\right]=\left[\frac{\left|\Delta d_i(z)B_i(z)\right|^+_{W_{n+1}(z)}}{W_n(z)}\right] \quad (44)$$

It is known that orthogonal bases of redundant PRNS are defined as

$$B_i(z)=m_i(z)\frac{W_{n+1}(z)}{p_i(z)}=m_i(z)\frac{W_n(z)p_{n+1}(z)}{p_i(z)} \quad (45)$$

Substitute Expression (45) into Equality (44). We obtain

$$S(z)=\left[\frac{\left|\Delta d_i(z)B_i(z)\right|^+_{W_{n+1}(z)}}{W_n(z)}\right]=\left[\frac{\Delta d_i(z)m_i(z)p_{k+1}(z)}{p_i(z)}\right] \quad (46)$$

The theorem is proved.

Example 3. Consider the application of Theorem 4 for the code of polynomial residue number system with informational bases $p_1(z)=z+1,\quad p_2(z)=z^2+z+1,$ $p_3(z)=z^4+z^3+z^2+z+1,\quad p_4(z)=z^4+z^3+1.$ The operating range of the code is $W_4(z)=\prod_{i=1}^{4}p_i(z)=z^{11}+z^8+z^7+z^5+z^3+z^2+z+1.$ Here, we select $p_5(z)=z^4+z+1$ as a control base.

Consider $D(z)=z^5=(1,z+1,1,z^3+z+1,z^2+z)$. The code combination for this polynomial is allowed, as $\deg D(z) < \deg W_4(z)$.

Let the error occur in the residue $d_1(z)$ and let the error depth be $\Delta d_1(z)=1$. Then, the erroneous combination is $D^*(z)=(0,z+1,1,z^3+z+1,z^2+z).$ Using Expression (41), the weight of the orthogonal basis is $m_1(z) = 1$; thus, we obtain

$$S(z)=\left[\frac{\Delta d_1(z)m_1(z)p_5(z)}{p_1(z)}\right]=\left[\frac{z^4+z+1}{z+1}\right]=z^3+z^2+z$$

Calculating the polynomial interval using the CRT for polynomials and using the orthogonal bases from Example 2, we have

$$D^*(z)=z^{14}+z^{13}+z^{12}+z^{11}+z^{10}+z^9+z^8+z^7+z^6+z^4+z^3+z^2+z+1$$

In this case, $S(z)=\left[\frac{D^*(z)}{W_n(z)}\right]=z^3+z^2+z$.

It is possible to identify both a location of the erroneous residue of the code combination and an error depth using Theorems 1–3. It is necessary to find the corresponding polynomial interval in order to do this. Due to the minimal introduced redundancy, a collision may occur in the code of PRNS with one control base. In this case, errors on informational bases and errors on control bases may produce the same value of the polynomial interval. Let the error occur in the residue $d_5(z)$ and let the error depth be $\Delta d_5(z)=z^2+z+1$. Then, we have $D^*(z)=(1,z+1,1,z^3+z+1,1)$. Using Expression (41), the weight of the orthogonal basis is $m_5(z) = z$; thus, we obtain

$$S(z) = \left[\frac{\Delta d_1(z) m_1(z) p_5(z)}{p_5(z)} \right] = \left[(z^2 + z + 1) z \right] = z^3 + z^2 + z$$

We have obtained a polynomial interval that coincides with the polynomial interval in the example. This is due to the fact that the number of possible polynomial intervals with one control base is equal to

$$N_1 = 2^{\deg p_{n+1}(z)} \tag{47}$$

At the same time, the number of polynomial intervals that correspond to errors on the bases of the code of PRNS is determined by

$$N_2 = \sum_{i=1}^{n+1} 2^{\deg p_i(z)} \tag{48}$$

It is obvious that $N_1 < N_2$. Therefore, codes of polynomial residue number system with one control base are only able to detect calculation errors. It is possible to eliminate this collision by increasing the introduced redundancy in PRNS.

3.2. Codes of Polynomial Residue Number System With Two Control Bases

Consider an ordered code of PRNS with informational bases $p_1(z), p_2(z), \ldots, p_n(z)$ for which the following expression takes place:

$$\deg p_1(z) \le \deg p_2(z) \le \ldots \le \deg p_n(z) \tag{49}$$

Now, we introduce two control bases $p_{n+1}(z), p_{n+2}(z)$ into this code that satisfy the following condition:

$$\deg p_{n+1}(z) + \deg p_{n+2}(z) \ge \deg p_n(z) + \deg p_{n-1}(z) \tag{50}$$

This will expand the full range of the code to

$$W_{n+2}(z) = \prod_{i=1}^{n+2} p_i(z) = W_n(z) p_{n+1}(z) p_{n+2}(z) = W_n(z) W_k(z) \tag{51}$$

where $W_k(z) = p_{n+1}(z) p_{n+2}(z)$.

An increase in redundancy in PRNS leads to an increase in the number of polynomial intervals. In this case,

$$N_1 = 2^{\deg(p_{n+1}(z) p_{n+2}(z))} \tag{52}$$

Here, the number of polynomial intervals that correspond to errors on bases of the code of PRNS is defined as

$$N_2 = \sum_{i=1}^{n+2} 2^{\deg p_i(z)} \tag{53}$$

The analysis of expressions shows that $N_1 > N_2$. Therefore, codes of polynomial residue number system with two control bases have the potential to correct single errors, which are understood as distortion of one residue in the code combination.

As codes of PRNS are modular codes, it is possible to use existing approaches to the development of algorithms designed to detect and correct errors; in other words, the procedure is the same as when codes of residue number system are used. It is proposed in [2–4] to use positional characteristics to construct error-correction codes of residue number system. It is known from [3] that the code combination of a number is allowed in codes of RNS if the following condition is met:

$$X = (x_1, ..., x_i, ..., x_{n+2}) < W_n \tag{54}$$

where $x_i \equiv X \bmod p_i$ and $W_n = \prod_{i=1}^{n} p_i$ is the operating range of the code of RNS.

At the same time, the full range of a redundant code of RNS with two control bases is

$$W_{n+2} = \prod_{i=1}^{n+2} p_i = W_n \, p_{n+1} p_{n+2} = W_n W_k \tag{55}$$

where $W_k = p_{n+1} p_{n+2}$.

The positional characteristic shows the location (position) of the number presented as the code combination of RNS relative to the operating range [2–4]. It is obvious that there are $W_k = p_{n+1} p_{n+2}$ operating ranges W_n within the full range W_{n+2}. If the numbers inside the full range are divided by the value W_n and the quotient of the result is taken, we obtain a positional characteristic, that is, the interval [5], equal to

$$S = \left[\frac{X}{W_n} \right] \tag{56}$$

If the code combination of RNS satisfies Expression (54), then the interval is equal to zero. If the number is outside the operating range, which corresponds to prohibited code combinations of RNS, then the value of the interval is in the range from 1 to $W_k - 1$. This means that the interval S depends on the modulus of the product of the control bases, W_k.

Through analysis of Expressions (43) and (56) we are able to conclude that there are similar approaches to the development of algorithms for detecting and correcting errors in codes of RNS and codes of polynomial residue number system. Hence, we can develop an algorithm for calculating the polynomial interval without using the division operation, as this operation is absent in the ring of irreducible polynomials.

Consider a code combination, $D(z) = (d_1(z), ..., d_i(z), ... d_{n+2}(z))$, presented in the code of PRNS with two control bases. Performing the inverse conversion using the CRT for polynomials,

$$D(z) = \sum_{i=1}^{n+2} d_i(z) B_i(z) - k(z) W_{n+2}(z) \tag{57}$$

where $k(z) = \left[\frac{d_1(z) B_1(z) + ... + d(z) B_i(z) + ... + d_{n+2}(z) B_{n+2}(z)}{W_{n+2}(z)} \right]$ is a rank of the polynomial.

Substituting this expression in (56), we obtain:

$$S(z) = \left[\frac{D(z)}{W_n(z)} \right] = \left[\frac{\sum_{i=1}^{n+2} d_i(z) B_i(z) - k(z) W_{n+2}(z)}{W_n(z)} \right] \quad (58)$$

The analysis in works [3–5] showed that both orthogonal bases of codes of RNS containing redundant bases and those without them have the similarity property. The conducted studies have confirmed this property for codes of polynomial residue number system. Thus, we have

$$\ddot{B}_i(z) \equiv B_i(z) \bmod W_n(z) \quad (59)$$

where $\ddot{B}_i(z)$ is an orthogonal basis of PRNS using only information bases $p_1(z), p_2(z), \ldots, p_n(z)$, and $B_i(z)$ is an orthogonal base of redundant PRNS.

Hence, orthogonal bases can be presented as

$$B_i(z) = W_n(z) L_i(z) + \ddot{B}_i(z) \quad (60)$$

where $L_i(z) = \left[\frac{B_i(z)}{W_n(z)} \right]$.

At the same time, for control bases we have $\ddot{B}_{n+1}(z) = 0, \ddot{B}_{n+2}(z) = 0$.

Then, using (47), we obtain

$$S(z) = \left[\frac{\sum_{i=1}^{n+2} d_i(z)(W_n(z) L_i(z) + \ddot{B}_i(z)) - k(z) W_{n+2}(z)}{W_n(z)} \right] = \left[\sum_{i=1}^{n+2} d_i(z) L_i(z) + \frac{\sum_{i=1}^{n} d_i(z) \ddot{B}_i(z)}{W_n(z)} - k(z) W_k(z) \right]. \quad (61)$$

When RNS was considered, it was shown that the values of the interval *S* form a ring modulo W_k. Then, based on the similarity of RNS and PRNS we can conclude that values of the polynomial interval $S(z)$ do not exceed values modulo $W_k(z)$ either. In this case, we obtain

$$S(z) = \left| \sum_{i=1}^{n+2} d_i(z) L_i(z) + \left[\frac{\sum_{i=1}^{n} d_i(z) \ddot{B}_i(z)}{W_n(z)} \right] - k(z) W_k(z) \right|_{W_k(z)} = \left| \sum_{i=1}^{n+2} d_i(z) L_i(z) + \ddot{k}(z) \right|_{W_k(z)}, \quad (62)$$

where $\ddot{\ddot{k}}(z)=\left[\frac{\sum_{i=1}^{n} d_i(z)\ddot{\ddot{B}}_i(z)}{W_n(z)}\right]$ is a rank of the polynomial in a non-redundant polynomial residue number system with bases $p_1(z), p_2(z),\ldots,p_n(z)$.

Thus, an algorithm has been developed for calculating the positional characteristic (a polynomial interval). Only modular operations are used in this algorithm, which allows us to detect and correct errors without performing a reverse conversion from the code of PRNS to a positional code or performing a division operation. Given that the calculations in PRNS are performed in parallel and independently on bases of the code, the error that occurred during the calculations on the *i*-th base will lead to a distortion of the final result only in the *i*-th residue. Therefore, the process of error detection and correction can be performed in parallel with conversion from the code of polynomial residue number system to the positional code, which reduces time costs.

Example 4. Using PRNS with three informational bases, $p_1(z)=z+1$, $p_2(z)=z^2+z+1$, and $p_3(z)=z^4+z^3+z^2+z+1$, and two control bases, $p_4(z)=z^4+z^3+1$ and $p_5(z)=z^4+z+1.$, the operating range of the code is $W_3(z)=\prod_{i=1}^{3} p_i(z)=z^7+z^6+z^5+z^2+z+1.$ The product of the two control bases is $W_k=p_{n+1}p_{n+2}=z^8+z^7+z^5+z^4+z^3+z+1.$ The full range of PRNS is $W_5(z)=\prod_{i=1} p_i(z)=z^{15}+1$.

As an allowed code combination of PRNS, we choose a polynomial, $D(z)=z^5+z^4+z+1=(0,\ z,\ z^3+z^2+1,\ 1,\ z^2+z)$, which satisfies Condition (32). The orthogonal bases of PRNS can be calculated as follows:

$$B_1(z)=z^{14}+z^{13}+z^{12}+z^{11}+z^{10}+z^9+z^8+z^7+z^6+z^5+z^4+z^3+z^2+z+1$$

$$B_2(z)=z^{14}+z^{13}+z^{11}+z^{10}+z^8+z^7+z^5+z^4+z^2+z$$

$$B_3(z)=z^{14}+z^{13}+z^{12}+z^{11}+z^9+z^8+z^7+z^6+z^4+z^3+z^2+z$$

$$B_4(z)=z^{14}+z^{13}+z^{12}+z^{11}+z^9+z^7+z^6+z^3$$

$$B_5(z)=z^{12}+z^9+z^8+z^6+z^4+z^3+z^2+z$$

The bases can then be divided into the operating range until the residue is obtained, thusly:

$$L_1(z)=z^7+z^4+z^2+z;\ \ddot{\ddot{B}}_1(z)=z^6+z^4+z^3+z^2+1$$

$$L_2(z)=z^7+z^5+z^2+z+1;\ \ddot{\ddot{B}}_2(z)=z^6+z^5+z+1$$

$$L_3(z)=z^7+z^4+z^3+z+1;\ \ddot{\ddot{B}}_3(z)=z^5+z^4+z^3+z^2+z+1$$

$$L_4(z) = z^7 + z^4 + z^3$$

$$L_5(z) = z^5 + z^4 + z$$

Calculate the polynomial interval of the code combination of PRNS.

Now, o obtain the value of the rank of the polynomial in non-redundant PRNS:

$$\dddot{k}(z) = \left[\frac{z(z^6 + z^5 + z + 1) + (z^3 + z^2 + 1)(z^5 + z^4 + z^3 + z^2 + z + 1)}{z^7 + z^6 + z^5 + z^2 + z + 1} \right] = z$$

This value is substituted in

$$S(z) = \left| \sum_{i=1}^{5} d_i(z) L_i(z) + \dddot{k}(z) \right|_{W_k(z)} = \left| z(z^7 + z^5 + z^2 + z + 1) + (z^3 + z^2 + 1) \times \right.$$

$$\left. \times (z^7 + z^4 + z^3 + z + 1) + (z^7 + z^4 + z^3) + (z^2 + z)(z^5 + z^4 + z) + z \right|_{W_k} =$$

$$= (z^{10} + z^9 + z^8 + z^6 + z^4 + z^2 + z + 1) \bmod z^8 + z^7 + z^5 + z^4 + z^3 + z + 1 = 0.$$

As the polynomial interval is equal to zero, it means that the code combination of PRNS does not contain an error.

Let the error occur on the base $p_1(z)$ and the error depth be $\Delta d_1(z) = 1$. Then, the code combination has the form

$$D^*(z) = (1^*, z, z^3 + z^2 + 1, 1, z^2 + z) =$$

$$= z^{14} + z^{13} + z^{12} + z^{11} + z^{10} + z^9 + z^8 + z^7 + z^6 + z^3 + z^2.$$

Using expression (56), we can find the polynomial interval:

$$S(z) = \left[\frac{z^{14} + z^{13} + z^{12} + z^{11} + z^{10} + z^9 + z^8 + z^7 + z^6 + z^3 + z^2}{z^8 + z^7 + z^5 + z^4 + z^3 + z + 1} \right] = z^7 + z^4 + z^2 + z.$$

We now use the developed algorithm. First, we obtain a rank of the polynomial in non-redundant PRNS, $\dddot{k}(z) = z$, that we substitute in

$$S(z) = \left| \sum_{i=1}^{5} d_i(z) L_i(z) + \dddot{k}(z) \right|_{W_k(z)} = \left| (z^7 + z^4 + z^2 + z) + z(z^7 + z^5 + z^2 + z + 1) + \right.$$

$$\left. + (z^3 + z^2 + 1)(z^7 + z^4 + z^3 + z + 1) + (z^7 + z^4 + z^3) + (z^2 + z)(z^5 + z^4 + z) + z \right|_{W_k} =$$

$$= (z^{10} + z^9 + z^8 + z^7 + z^6 + 1) \bmod z^8 + z^7 + z^5 + z^4 + z^3 + z + 1 = z^7 + z^4 + z^2 + z.$$

The obtained results coincide, which indicates the possibility of using redundant codes of polynomial residue number system to build fault-tolerant high-speed computing systems for digital signal processing and cryptographic protection of information operating in Galois fields.

4. Conclusions

The article discusses the principles of constructing redundant codes of polynomial residue number system. Several theorems concerning the properties of codes of PRNS were considered and proven. It was shown that the distortion of any residue transforms an allowed code combination into a prohibited one. In this case, the criterion for the presence of an error in the code combination of PRNS is a violation of Condition (32). Studies of the code of polynomial residue number system, which has minimal redundancy, have been carried out. We have shown that this code only allows for detection an error in the code combination of PRNS. It was proposed to introduce two control bases to expand

error-correction abilities. We have shown that the use of two control bases makes it possible to correct any erroneous residue in the code combination of PRNS. The article presents the developed algorithm for calculating the positional characteristic (a polynomial interval) in which only modular operations are used. This algorithm makes it possible to detect and correct errors without performing a reverse conversion from a code of PRNS to a positional code or performing a division operation, which is confirmed by the examples given in the article.

Author Contributions: Conceptualization, I.A.K.; Data curation, K.T.T., A.A.O. and N.K.C.; Formal analysis, I.A.K. and V.P.P.; Investigation, I.A.K., V.P.P., K.T.T. and A.A.O.; Methodology, I.A.K.; Project administration, I.A.K.; Software, N.K.C.; Supervision, I.A.K.; Validation, V.P.P., K.T.T. and A.A.O.; Visualization, V.P.P., K.T.T. and A.A.O.; Writing—original draft, I.A.K.; Writing—review and editing, N.K.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Russian Foundation for Basic Research, project No. 20-37-90009 and by the Russian Science Foundation (Moscow), project No. 22-21-00768.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

References

1. Szabo, N.S.; Tanaka, R.I. *Residue Arithmetic and Its Applications to Computer Technology*; Mc-Graw Hill: New York, NY, USA, 1967.
2. Mohan, P.V. *Residue Number Systems. Algorithms and Architectures*; Publisher: Springer, New York, NY, USA, 2002.
3. Omondi, A.; Premkumar, B. *Residue Number Systems: Theory and Implementation*; Publisher: Imperial College, London, UK, 2007.
4. Mohan, A. *Residue Number Systems. Theory and Applications*; Springer International Publishing: Cham, Switzerland, 2016.
5. Vergos, H.T.; Efstathiou, C. A unifying approach for weighted and diminished-1 modulo ($2^n + 1$) addition. *IEEE Trans. Circuits Syst. II Express Briefs* **2008**, *55*, 1041–1045.
6. Patel, R.A.; Benaissa, M.; Boussakta, S. Fast Parallel-Prefix Architectures for Modulo 2n − 1 Addition with a Single Representation of Zero. *IEEE Trans. Comput.* **2007**, *56*, 1484–1492.
7. Juang, T.B.; Chiu, C.-C.; Tsai, M.-Y. Improved Area-Efficient Weighted Modulo 2^n+1 Adder Design With Simple Correction Schemes. *IEEE Trans. Circuits Syst. II Express Briefs* **2010**, *57*, 198–202.
8. Bajard, J.-C.; Eynard, J.; Gandino, F. Fault Detection in RNS Montgomery Modular Multiplication. In Proceedings of the 2013 IEEE 21st Symposium on Computer Arithmetic, Austin, TX, USA, 7–10 April **2013**. https://doi.org/10.1109/ARITH.2013.31.
9. Gandino, F.; Lamberti, F.; Paravati, G.; Bajard, J.-C.; Montuschi, P. An Algorithmic and Architectural Study on Montgomery Exponentiation in RNS. *IEEE Trans. Comput.* **2012**, *61*, 1071–1083.
10. Plantard, T. Efficient Word Size Modular Arithmetic. *IEEE Trans. Emerg. Top. Comput.* **2021**, *9*, 1506–1518.
11. Shahana, T.K.; Jose, B.R.; James, R.K.; Jacob, K.P.; Sasi, S. RRNS-Convolutional encoded concatenated code for OFDM based wireless communication. In Proceedings of the 2008 16th IEEE International Conference on Networks, New Delhi, India, 12–14 December 2008. https://doi.org/10.1109/ICON.2008.4772571.
12. Albicocco, P.; Cardarilli, G.C.; Nannarelli, A.; Re, M. Twenty years of research on RNS for DSP: Lessons learned and future perspectives. In Proceedings of the 2014 International Symposium on Integrated Circuits (ISIC), Singapore, 10–12 December 2014. https://doi.org/10.1109/ISICIR.2014.7029575.
13. Khalifa, M.A.; Emam, A.E.; Youssef, M.I. A Comparative Study for RNS Coding Scheme Performance in OFDM and SC-FDMA Systems. In Proceedings of the 2020 12th International Conference on Electrical Engineering (ICEENG), Cairo, Egypt, 7–9 July 2020. https://doi.org/10.1109/ICEENG45378.2020.9171754.
14. Ananthalakshmi, A.V.; Rajagopalan, P. VLSI implementation of residue number system based efficient digital signal processor architecture for wireless sensor nodes. *Int. J. Inf. Technol.* **2019**, *11*, 829–840.
15. Cardarilli, G.; Nunzio, L.D.; Fazzolari, R.; Nannarelli, A.; Petricca, M.; Re, M. Design Space Exploration based Methodology for Residue Number System Digital Filters Implementation. *IEEE Trans. Emerg. Top. Comput.* **2020**, *10*, 186–198.
16. Praveen, M.; Reddy, M.V.S.R.; Raju, Y.D. S. An approach for fixed coefficient RNS-based FIR filter. *Open Access Int. J. Sci. Eng.* **2021**, *6*, 1–6.
17. Luan, Z.; Chen, X.; Ge, N.; Wang, Z. Simplified fault-tolerant FIR filter architecture based on redundant residue number system. *Electron. Lett.* **2014**, *50*, 1768–1770.
18. Bernocchi, G.L.; Cardarilli, G.C.; Re, A.D.; Nannarelli, A.; Re, M. Low-power adaptive filter based on RNS components. In Proceedings of the 2007 IEEE International Symposium on Circuits and Systems, New Orleans, LA, USA, 27–30 May 2007. https://doi.org/10.1109/ISCAS.2007.378155.
19. Rajni, D.S.K. Efficient Techniques for FIR Filter Designing. *Proc. First Int. Conf. Comput. Electron. Wirel. Commun.* **2022**, *329*, 97–114.

20. Rawat, P.; Nemade, S. High Speed and Area Efficient Pipelined Distributed Arithmetic Technique based Adaptive Filter. *Int. J. Electron. Commun. Comput. Eng.* **2020**, *11*, 8–15.
21. Kaplun, D.; Voznesensky, A.; Veligosha, A.V.; Kalmykov, I.A.; Sarma, K.K. Technique to Adjust Adaptive Digital Filter Coefficients in Residue Number System Based Filters. *IEEE Access* **2021**, *9*, 82402–82416.
22. Olsen, E.B. RNS Hardware Matrix Multiplier for High Precision Neural Network Acceleration: "RNS TPU". In Proceedings of the 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, 27–30 May 2018. https://doi.org/10.1109/ISCAS.2018.8351352.
23. Nakahara, H.; Sasao, T. A High-speed Low-power Deep Neural Network on an FPGA based on the Nested RNS: Applied to an Object Detector. In Proceedings of the 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, 27–30 May 2018. https://doi.org/10.1109/ISCAS.2018.8351850.
24. Salamat, S.; Imani, M.; Gupta, S.; Rosing, T. RNSnet: In-Memory Neural Network Acceleration Using Residue Number System. In Proceedings of the 2018 IEEE International Conference on Rebooting Computing (ICRC), McLean, VA, USA, 7–9 November 2018. https://doi.org/10.1109/ICRC.2018.8638592.
25. Ciet, M.; Neve, M.; Peeters, E.; Quisquater, J.-J. Parallel FPGA implementation of RSA with residue number systems—Can side-channel threats be avoided? In Proceedings of the 2003 46th Midwest Symposium on Circuits and Systems, Cairo, Egypt, 27–30 December 2003. https://doi.org/10.1109/MWSCAS.2003.1562409.
26. Papachristodoulou, L.; Fournaris, A.P.; Papagiannopoulos, K.; Batina, L. Practical Evaluation of Protected Residue Number System Scalar Multiplication. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2018**, *2019*, 259–282.
27. Noordam, L. VHDL Implementation of 4096-bit RNS Montgomery Modular Exponentiation for RSA Encryption. *TU Delft Electr. Eng. Math. Comput. Sci.* **2019**.
28. Wu, T. WITHDRAWN: Fast RNS elliptic curve point multiplication on FPGAs. *Microprocess. Microsyst.* **2020**. https://doi.org/10.1016/j.micpro.2020.103442.
29. Leelavathi, G.; Shaila, K.; Venugopal, K.R. Elliptic Curve Cryptography implementation on FPGA using Montgomery multiplication for equal key and data size over GF(2^m) for Wireless Sensor Networks. In Proceedings of the 2016 IEEE Region 10 Conference (TENCON), Singapore, 22–25 November 2016. https://doi.org/10.1109/TENCON.2016.7848043.
30. Agbedemnab, P.A.; Agebure, M.; Akobre, S. A Fault Tolerant Scheme for Detecting Overflow in Residue Number Microprocessors. *Int. J. Eng. Comput. Sci.* **2018**, *7*, 23578–23587.
31. Tay, T.F.; Chang, C.-H. Fault-Tolerant Computing in Redundant Residue Number System. In *Embedded Systems Design with Special Arithmetic and Number Systems*; Springer: Berlin/Heidelberg, Germany, **2017**; pp. 65–88.
32. Popov, D.I.; Gapochkin, A.V. Development of Algorithm for Control and Correction of Errors of Digital Signals, Represented in System of Residual Classes. In Proceedings of the 2018 International Russian Automation Conference (RusAutoCon), Sochi, Russia, 9–16 September 2018. https://doi.org/10.1109/RUSAUTOCON.2018.8501826.
33. Afriyie, Y. A Novel Exploitation of Errors in Redundant Residue Number System Architecture. *Am. J. Appl. Sci.* **2021**, *18*, 96–106.
34. Selivanova, M. V.; Tyncherov, K. T.; Ikhsanova, F. A.; Kalmykov, I.A.; Olenev, A. A. A method of paired zeroing of numbers in a residue system. *Journal of Physics: Conference Series*. **2019**, Issue 2, Volume 1333. https://doi.org/10.1088/1742-6596/1333/2/022015.
35. Pontarelli, S.; Cardarilli, G.C.; Re, M.; Salsano, A. A Novel Error Detection and Correction Technique for RNS Based FIR Filters. In Proceedings of the 2008 IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems, Cambridge, MA, USA, 1–3 October 2008. Available online: https://doi.org/10.1109/DFT.2008.32 (accessed on 17 January 2021).
36. Gao, Z.; Reviriego, P.; Pan, W.; Xu, Z.; Zhao, M.; Wang, J.; Maestro, J.A. Fault Tolerant Parallel Filters Based on Error Correction Codes. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2015**, *23*, 384–387.
37. Kalmykov, I.A.; Pashintsev, V.P.; Zhuk, A.P.; Kalmykov, M.I.; Olenev, A.A. Redundant Modular Codes for Development of Fault-Tolerant Systems of Satellite Identification. *Int. J. Emerg. Trends Eng. Res.* **2020**, *8*, 3160–3168.
38. Sachenko, A.; Zhengbing, H.;Yatskiv, V. Increasing the Data Transmission Robustness in Wsn Using the Modified Error Correction Codes on Residue Number System. *Elektron. Ir Elektrotechnika* **2015**, *21*, 76–81.
39. Tay, T.F.; Chang, C.-H. A new algorithm for single residue digit error correction in Redundant Residue Number System. In Proceedings of the 2014 IEEE International Symposium on Circuits and Systems (ISCAS), Melbourne, VIC, Australia, 1–5 June 2014. https://doi.org/10.1109/ISCAS.2014.6865493.
40. Tang, Y.; Boutillon, E.; Jégo, C.; Jézéquel, M. A new single-error correction scheme based on self-diagnosis residue number arithmetic. In Proceedings of the 2010 Conference on Design and Architectures for Signal and Image Processing (DASIP), Edinburgh, UK, 26–28 October 2010. https://doi.org/10.1109/DASIP.2010.5706242.
41. Alhassan, A.; Saeed, I.; Agbedemnab, P.A. The Huffman's Method of Secured Data Encoding and Error Correction using Residue Number System (RNS). *Commun. Appl. Electron. (CAE)* **2015**, *2*, 14–18.
42. Zhu, D.; Natarajan, B. Residue number system arithmetic inspired hopping pilot pattern design for cellular downlink OFDMA. In Proceedings of the 2010 Wireless Telecommunications Symposium (WTS), Tampa, FL, USA, 21–23 April 2010. https://doi.org/10.1109/WTS.2010.5479666.
43. Timarchi, S.; Fazlali, M. Generalised fault-tolerant stored-unibit-transfer residue number system multiplier for moduli set { 2^n–1, 2^n, 2^n+1}. *IET Comput. Digit. Tech.* **2012**, *6*, 269–276.

44. Afriyie, Y.; Daabo, M.I. Multiple Bits Error Detection and Correction in RRNS Architecture using the MRC and HD Techniques. *Int. J. Comput. Appl.* **2018**, *180*, 18–23.
45. James, J.; Pe, A. Error correction based on redundant Residue Number System. In Proceedings of the 2015 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, 10–11 July 2015. https://doi.org/10.1109/CONECCT.2015.7383940.
46. Skavantzos, A.; Taylor, F.J. On the polynomial residue number system (digital signal processing). *IEEE Trans. Signal Processing* **1991**, *39*, 376–382.
47. Skavantzos, A.; Stouraitis, T. Polynomial residue complex signal processing. *IEEE Trans. Circuits Syst. II Analog. Digit. Signal Processing* **1993**, *40*, 342–344.
48. Yang, M.-C.; Wu, J.-L. A new interpretation of "polynomial residue number system". *IEEE Trans. Signal Processing* **1994**, *42*, 2190–2191.
49. Paliouras, V.; Skavantzos, A.; Stouraitis, T. Multi-voltage low power convolvers using the polynomial residue number system. In *GLSVLSI '02: Proceedings of the 12th ACM Great Lakes symposium on VLSI*; 18 April 2002; pp. 7–11. https://doi.org/10.1145/505306.505309.
50. Abdallah, M.; Skavantzov, A. The multipolynomial channel polynomial residue arithmetic system. *IEEE Trans. Circuits Syst. II Analog. Digit. Signal Processing* **1999**, *46*, 165–171.
51. Katkov, K.A.; Makarova, A.V.; Parallel modular technologies in digital signal processing. *Life Sci. J.* **2014**, *11*, 435–438.
52. Chu, J.; Benaissa, M. GF(2^m) multiplier using Polynomial Residue Number System. In Proceedings of the APCCAS 2008—2008 IEEE Asia Pacific Conference on Circuits and Systems, Macao, China, 30 November–3 December 2008. https://doi.org/10.1109/APCCAS.2008.4746320.
53. Kalimoldayev, M.; Tynymbaev, S.; Magzom, M.; Ibraimov, M.; Khokhlov, S.; Abisheva, A.; Sydorenko, V. Polynomials Multiplier under Irreducible Polynomial Module for High-Performance Cryptographic Hardware Tools. Available online: http://ceur-ws.org/Vol-2393/paper_363.pdf (accessed on 15 January 2021).
54. Kalimoldayev, M.; Tynymbayev, S.; Gnatyuk, S.; Khokhlov, S.; Magzom, M.; Kozhagulov, Y. Matrix multiplier of polynomials modulo analysis starting with the lower order digits of the multiplier. *News Natl. Acad. Sci. Repub. Kazakhstan Ser. Geol. Technol. Sci.* **2019**, *4*, 181–187.
55. Kalimoldayev, M.; Tynymbayev, S.; Gnatyuk, S.; Ibraimov, M.; Magzom, M. The device for multiplying polynomials modulo an irreducible polynomial. *News Natl. Acad. Sci. Repub. Kazakhstan Ser. Geol. Technol. Sci.* **2019**, *2*, 199–205.
56. Biyashev, R.G.; Nyssanbayeva, S.E.; Begimbayeva, Y.Y.; Magzom, M.M. Modification of the cryptographic algorithms, developed on the basis of nonpositional polynomial notations. Proceedings of the International Conference on Circuits, Systems, Signal Processing, Communications and Computers (CSSCC 2015). **2015**, pp. 170–176.
57. Kapalova, N.; Dyusenbayev, D. Security analysis of an encryption scheme based on nonpositional polynomial notations. *Open Eng.* **2016**, *6*, 250–258.
58. Biyashev, R.; Nyssanbayeva, S.; Kapalova, N. Secret keys for nonpositional cryptosystems. Development, investigation and implementation. *Lambert Acad. Publ.* **2014**, *136-147 p*.
59. Good, T.; Benaissa, M. Very small FPGA application-specific instruction processor for AES. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2006**, *53*, 1477–1486.
60. Mangard, S.; Aigner, M.; Dominikus, S. A highly regular and scalable AES hardware architecture. *IEEE Trans. Comput.* **2003**, *52*, 483–491.
61. Hodjat, A.; Verbauwhede, I. Minimum area cost for a 30 to 70 Gbits/s AES processor. *IEEE Comput. Soc. Annu. Symp. VLSI* **2004**, 83–88. https://doi.org/10.1109/ISVLSI.2004.1339512.
62. Dichenko, S.A.; Finko, O.A. Controlling and Restoring the Integrity of Multi-Dimensional Data Arrays through Cryptocode Constructs. *Program. Comput. Softw.* **2021**, *47*, 415–425.
63. Samoylenko, D.V.; Eremeev, M.A.; Finko, O.A. A method of providing the integrity of information in the group of robotic engineering complexes based on crypt-code constructions. *Autom. Control. Comput. Sci.* **2017**, *51*, 965–971.
64. Martínez-Peñas, U. Sum-Rank BCH Codes and Cyclic-Skew-Cyclic Codes. *IEEE Trans. Inf. Theory* **2021**, *67*, 5149–5167.
65. Li, C.; Wu, P.; Liu, F. On Two Classes of Primitive BCH Codes and Some Related Codes. *IEEE Trans. Inf. Theory* **2019**, *65*, 3830–3840.