CONTROL AND CORRECTION OF DATA ERRORS IN A RESIDUE CLASS

Abstract. The subject of the research in the article is the methods of control and correction of single errors of integer data are presented in the residue class (RC), which allows to increase the efficiency of using RC when building computer systems and components. The purpose of the article is to develop a method for correcting single data errors in RC. Tasks: to study the code structures presented in RC to determine the possibility of control and correction of data errors; to investigate the effect of RC properties on data control and correction operations; to develop a method for correcting single data errors in RC. Research methods: methods of analysis and synthesis of computer systems, number theory, coding theory in RC. The following results are obtained. An analysis of the correcting capabilities of codes in RC showed the high efficiency of using non-positional code structures, which is due to the presence of primary and secondary redundancy in such structures. The article presents a method for correcting one-time data errors in RC. Examples of detecting and correcting data errors in RC code are given, which confirms the theoretical results obtained. Conclusions. Studies have shown that the use of codes in RC makes it possible to build an effective system for monitoring and correcting data errors with the introduction of minimal code redundancy. That is, when certain conditions are met, the introduction of one control base allows not only monitoring, but also correction of single data errors.

Keywords: non-positional code structure, residue classes, positional numeral systems, minimum code distance, error-control coding, data control and correction.

Introduction

In general, in order to verify, diagnose and correct errors a code structure requires a certain error-correcting capability. In this case, code is required to be introduced to data duplication, i.e. information redundancy should be implemented. All of the above fully refers to a non-positional code structure (NCS) in residue classes (RC) [1-3]. For each random RC the amount of redundancy \( R = M_0 / M \) uniquely determines correction capability of a non-positional error-correcting code. Error correcting codes in RC can have any given values of minimum code distance (MCD) \( d_{\text{min}}^{(\text{RC})} \), which depends on the value of redundancy \( R \). The acquainted theorem [1-2] establishes a link between error-correcting code redundancy \( R \), the value of MCD \( d_{\text{min}}^{(\text{RC})} \), and the amount of RC check bases \( k \).

Error-correcting code has MCD values \( d_{\text{min}}^{(\text{RC})} \) in case when the degree of redundancy \( R \) is not less than the product \( d_{\text{min}}^{(\text{RC})} - 1 \) of RC bases.

On the one hand we get

\[
R \geq \prod_{i=1}^{d_{\text{min}}^{(\text{RC})} - 1} m_i,
\]

but on the other hand

\[
R = M_0 / M = \prod_{i=1}^{n+k} m_i / \prod_{i=1}^{n} m_i = \prod_{i=1}^{k} m_{n+i}.
\]

In this case, it’s correct to state that

\[
d_{\text{min}}^{(\text{RC})} - 1 = k,
\]

or

\[
d_{\text{min}}^{(\text{RC})} = k + 1. \tag{1}
\]

There are two approaches to solve the problem of providing NCS with all required error-correcting properties in RC.

The first approach. If the requirements for error-correcting properties of NCS are known, for example, depending on amount of errors being detected \( t_{\text{det}} \) or corrected \( t_{\text{cor}} \) required information redundancy \( R \) should be introduced, using the amount of \( k \) or the value \( \{m_i\} \) of check bases. Redundancy \( R \) determines minimum code distance \( d_{\text{min}}^{(\text{RC})} \) of NCS in RC. Then, according to the error-control coding (ECC) theory for an ordered \((m_i < m_{i+1})\) RC we have that

\[
t_{\text{det}} \leq d_{\text{min}}^{(\text{RC})} - 1, \tag{2}
\]

\[
t_{\text{det}} \leq k; \tag{3}
\]

\[
t_{\text{cor}} \leq \left[ (d_{\text{min}}^{(\text{RC})} - 1)/2 \right], \tag{4}
\]

\[
t_{\text{cor}} \leq [k/2]. \tag{5}
\]

The second approach. For a given NCS

\[
A_{\text{RC}} = \{a_n, a_{n-k}, \ldots, a_{i+1}, a_i, a_{i-1}, \ldots, a_1, a_2, \ldots, a_n\} \tag{6}
\]

(for a given value \( k \)) its error-correcting capabilities (determined by the \( d_{\text{min}}^{(\text{RC})} \) value) of RC code are defined by the expressions (3) and (5).

Note that, if an ordered RC is extended by adding \( k \) check bases to \( n \) information modules, then MCD \( d_{\text{min}}^{(\text{RC})} \) of the error-correcting code is increased by the value \( k \) (see expression (1)).

The values of \( d_{\text{min}}^{(\text{RC})} \) can be also increased by decreasing the number \( n \) of information bases, i.e. by transitioning to less accurate calculations. It’s clear that in RC between error-correcting \( R \) properties of error-control codes and calculation accuracy \( W \) inverse proportion exists. The same computer can perform arithmetical calculations or any other math operations both
with high \( W \) accuracy but a low error-correcting \( R \) capability and with lower \( W \) accuracy, but with a higher capability \( R \) of error detection and correction in order to verify, diagnose and correct data faults, as well as to demonstrate higher data processing performance (the time to execute basic operations is inversely proportional to \( n \) information bases in RC) [2-4].

The purpose of the article is to develop a method for correcting single data errors in RC.

The main part

Now we’ll analyze the process of single-error correcting redundancy in RC given the minimal information redundancy by introduction of a single \((k = 1)\) check base.

In this case, according to the error control coding theory in RC [1, 5], MCD is equal to the value

\[
d_{\min}^{(RC)} = k + 1.\]

If \( k = 1 \), then MCD is \( d_{\min}^{(RC)} = 2 \), which, as according to the general error control coding theory, ensures any single-error detection (an error in one of the residues \( a_i \) \((i = 1, n + 1)\) ) in NCS.

In general, just as in the positional numeral system (PNS), the process of data error correction in RC consists of three stages.

The first stage - data checking (correctness or incorrectness verification of the initial number \( A_{RC} \)). On the second stage diagnosing the false \( \bar{A}_{RC} \) number (detection of a single corrupted residue \( \bar{a}_i \) of the number \( A_{RC} \) to the base \( m_i \) in RC). And, finally, on the third stage correcting the invalid residue \( \bar{a}_i \) to its true value \( a_i \) of the number, i.e. correcting false \( \bar{A}_{RC} \) number (getting the correct number \( A_{RC} = \bar{A}_{cor} \)).

The degree of information redundancy \( R \) (code error-correcting property) is estimated by the value of MCD \( d_{\min}^{(PNS)} \). As previously noted, the value of MCD is defined by the ratio \( d_{\min}^{(RC)} = k + 1 \), where \( k \) is the amount of check bases in an ordered RC.

Let’s start with the NCS

\[
A_{RC} = (a_1 || a_2 || ... || a_{i-1} || a_i || a_{i+1} || ... || a_n || ... || a_{n+k})
\]

in RC having a minimal \((k = 1)\) additional information redundancy. In this situation it’s considered that

\[
d_{\min}^{(RC)} = 2.\]

According to the error control coding theory in PNS if the minimum code distance is granted to be

\[
d_{\min}^{(PNS)} = 2,
\]
a single error in a code structure is ensured to be detected. In PNS a single error is understood as a corruption of a single information bit, for instance \( 0 \rightarrow 1 \) or \( 1 \rightarrow 0 \). In order to correct this single error it’s required to ensure the condition, when

\[
d_{\min}^{(PNS)} = 3.
\]

Contrary to PNS, a single error in RC is understood as a corruption of a single residue \( a_i \) modulo \( m_i \). Inasmuch as the residue \( a_i \) of the number \( A_{RC} \) modulo \( m_i \) contains \( z = \{\log_2(m_i - 1)\} + 1 \) binary bits, then it’s formally correct to be considered that if

\[
d_{\min}^{(RC)} = 2 \quad (k = 1)
\]
is within limits of a single residue \( a_i \), an error cluster can be detected in RC, with its length not exceeding \( z \) binary bits. However, in RC, as it is shown in literature [1, 2, 6], there are some cases when a single error can be corrected while

\[
d_{\min}^{(RC)} = 2.
\]

In the light of specific features and properties of NCS representation in RC an error-correcting capability given

\[
d_{\min}^{(RC)} = 2
\]
can be explained in the following manner.

1. A single error in PNS and in RC are different concepts, as it was shown before. With that being said, MCD \( d_{\min}^{(PNS)} \) for PNS and \( d_{\min}^{(RC)} \) for RC has different meaning and measure.

2. Existing (implicitly) intrinsic (natural, primal) information redundancy in NCS, being stored in residues \( \{a_i\} \) due to their forming procedure, has a positive effect (from the perspective of increasing data jam-resistance, transfer and processing reliability) that kicks in only with the presence of a subsidiary (artificial, secondary) information redundancy.

An artificial information redundancy in NCS is being introduced by using (additionally to \( n \) information bases) \( k \) check bases in RC. A distinguishing feature of RC is its significant display of the intrinsic information redundancy only if the subsidiary one is also present, due to introduction of check bases.

3. As shown in [6-8], error control code in RC with mutually prime bases has the MCD value of \( d_{\min}^{(RC)} \) only if the information redundancy level is not less than the product of any \( d_{\min}^{(RC)} - 1 \) bases of a given RC.

The availability and interaction of primary and secondary redundancies during the subsidiary tests (time redundancy usage) of error-correcting process, which may provide a single-error error-correcting capability in RC, while \( d_{\min}^{(RC)} = 2 \) (given \( k = 1 \)).

Indeed, according to the expressions (3) and (5) for an ordered RC following conclusions can be made: with a single \((k = 1)\) check base \( m_{n+1} \) in RC, the NCS

\[
A = (a_1 || a_2 || ... || a_{i-1} || a_i || a_{i+1} || ... || a_n || a_{n+1})
\]

can have several values of \( d_{\min}^{(RC)} \). In this case, it depends on the value of check residue \( m_{n+1} \). If, for every
different RC modulus condition \( m_i < m_{n+1} \) \((i = \frac{1}{3}, n)\) is met, then conclusion can be made that \( d^{(RC)}_{\text{min}} = 2 \), as according to the expression (1), and \( t_{\text{det}} = 1 \), according to the expression (2). If the condition \( m_i, m_j < m_{n+1} \) \((i, j = \frac{1}{3}, n ; i \neq j)\) is met across the totality of \(\{ m_i \}\) information bases for a random modulus pair, then
\[ d^{(RC)}_{\text{min}} = 3 \quad \text{and} \quad t_{\text{det}} = 2. \]

Thus, for the NCS in RC given \( k = 1 \), the MCD \( d^{(RC)}_{\text{min}} \) can vary, depending on the value of RC check base \( m_{n+1} \). Assume, RC is given information bases \( m_1 = 3 \), \( m_2 = 4 \), \( m_3 = 5 \), \( m_4 = 7 \) and moreover \( m_k = m_{n+1} = m_5 = 11 \). In this case error verification of any single corrupted NCS residue can be ensured.

Number representation specificity in RC in some cases allows not only to detect an error, but to find a place of its occurrence with the use of a single check base, which would be impossible to do in the PNS, utilizing existing methods of detecting and correcting errors.

Let’s assume, that in the corrupted \( \tilde{A} ≥ M \) number
\[ \tilde{A} = (a_1 || a_2 || \ldots || a_{i-1} || \tilde{a}_i || a_{i+1} || \ldots || a_n || a_{n+1}) \]
the error
\[ \tilde{a}_i = (a_i + \Delta a_i) \mod m_i \]
is verified to be present in the residue \( a_i \) modulo \( m_i \).

We’ll take a look at the ratio, which makes it possible to correct an error in a given residue \( \tilde{a}_i \) [1].

It’s clear that:
\[ \tilde{A} = (A + \Delta A) \mod M_0 \]

Basing on that the error magnitude can be equated to
\[ \Delta A = (0 || 0 || \ldots || 0 || \Delta a_i || 0 || \ldots || 0 || 0), \]
then the correct \( A < M \) number \( A \) can be expressed as follows:
\[ A = \tilde{A} - \Delta A \mod M_0 = \]
\[ = [a_1 || a_2 || \ldots || a_{i-1} || \tilde{a}_i || a_{i+1} || \ldots || a_n || a_{n+1}] - \]
\[ - (0 || 0 || \ldots || 0 || \Delta a_i || 0 || \ldots || 0 || 0) \mod M_0 = \]
\[ = [a_1 || a_2 || \ldots || a_{i-1} || (\tilde{a}_i - \Delta a_i) \mod m_i || a_{i+1} || \ldots || a_n || a_{n+1}] \mod M_0. \]

We’ll quantify the value of \( A \). Inasmuch number \( A \) is correct, i.e. contained in numerical interval \([0, M]\), then the following inequality will be fulfilled:
\[ A = (\tilde{A} - \Delta A) \mod M_0 < M. \]

Basing on the value of the error \( \Delta A \) is equal to
\[ \Delta A = \Delta a_i \cdot B_j, \]
then the inequality (7) will be expressed as:
\[ \tilde{A} - \Delta a_i \cdot B_j - r \cdot M_0 < M \quad \text{or} \]
\[ \tilde{A} - \Delta a_i \cdot B_j - r \cdot M_0 < M \quad \text{and} \]
\[ \tilde{A} - (\tilde{a}_i - \Delta a_i) \cdot B_j - r \cdot M_0 < M \]
\[ a_i - \tilde{a}_i < (M_0 / m_{n+1}) / B_j - \tilde{A} / B_j + r \cdot M_0 / B_j. \]

Since the orthogonal base of RC module \( m_i \) takes the form of \( B_j = m_i \cdot M_0 / m_j \), then the expression (8) shows up as:
\[ a_i - \tilde{a}_i + (m_i + r \cdot m_i \cdot m_{n+1}) / (m_i \cdot m_{n+1}) - \tilde{A} / B_j \quad \text{or} \]
\[ a_i - \tilde{a}_i + m_i (1 + r \cdot m_{n+1}) / (m_i \cdot m_{n+1}) - \tilde{A} / B_j. \]

Inasmuch as the value of the residue \( a_i \) is a natural number, then the value of
\[ m_i (1 + r \cdot m_{n+1}) / (m_i \cdot m_{n+1}) - \tilde{A} / B_j, \]
as shown in the expression (9), should be an integer.

Thus, taking an integral part of the last ratio, the formula for correcting error in the residue \( \tilde{a}_i \) of the number \( \tilde{A} \) will be:
\[ a_i = (\tilde{a}_i + [m_i (1 + r \cdot m_{n+1}) / (m_i \cdot m_{n+1}) - \tilde{A} / B_j] \mod m_i). \]

We’ll have a look at the examples of error correction in RC.

Example №1. Perform data verification of the number
\[ A_{RC} = (0 || 0 || 0 || 0 || 5) \]
and correct it if required, when RC was given information \( m_1 = 3 \), \( m_2 = 4 \), \( m_3 = 5 \), \( m_5 = 7 \) and check \( m_k = m_{n+1} = 11 \) bases.

Thereby,
\[ M = \prod_{i=1}^{n} m_i = \prod_{i=1}^{4} m_i = 420 \]
and
\[ M_0 = M \cdot m_{n+1} = 420 \cdot 11 = 4620. \]

Orthogonal RC bases \( B_i \) \((i = 1, n+1)\) and their weights are equal
\[ B_1 = (1 || 0 || 0 || 0 || 0) = 1540, \quad \bar{m}_1 = 1; \]
\[ B_2 = (0 || 1 || 0 || 0 || 0) = 3465, \quad \bar{m}_2 = 3; \]
\[ B_3 = (0 || 0 || 1 || 0 || 0) = 3696, \quad \bar{m}_3 = 4; \]
The set of and the totality of the quotient $B_j$ orthogonal RC bases are shown in Table 1 respectively.

\[
\begin{array}{c|cccc}
   \hline
   B_j & 1 & 2 & 3 & 4 \\
   \hline
   1 & 385 & 616 & 1100 & 980 \\
   2 & 385 & 231 & 330 & 210 \\
   3 & 616 & 693 & 792 & 672 \\
   4 & 220 & 165 & 396 & 540 \\
   5 & 280 & 105 & 336 & 120 \\
   \hline
\end{array}
\]

Now then (Table 1):

\[
\hat{A}_{jPNS} = \left( \sum_{i=1}^{n} a_i \cdot B_{ji} \right) \mod M_j = (0 \cdot 385 + 0 \cdot 616 +
0 \cdot 1100 + 5 \cdot 980) \mod 1540 = 280 < 420 .
\]

Arriving at conclusion, that the residue $a_1$ of the number $\hat{A}_1$ is possibly a corrupted residue $\overline{a}_1$;

\[
\hat{A}_{2PNS} = \left( \sum_{i=1}^{n} a_i \cdot B_{j2} \right) \mod M_2 = (0 \cdot 385 + 0 \cdot 231 +
0 \cdot 330 + 5 \cdot 210) \mod 1155 = 1050 > 420 .
\]

Hence, the residue $a_2$ is ensured being not corrupted;

\[
\hat{A}_{3PNS} = \left( \sum_{i=1}^{n} a_i \cdot B_{j3} \right) \mod M_3 = (0 \cdot 616 + 0 \cdot 693 +
0 \cdot 792 + 5 \cdot 672) \mod 924 = 588 < 420 .
\]

Deduced, the residue $a_3$ is ensured being not corrupted;

\[
\hat{A}_{4PNS} = \left( \sum_{i=1}^{n} a_i \cdot B_{j4} \right) \mod M_4 =
(0 \cdot 220 + 0 \cdot 165 + 0 \cdot 369 + 5 \cdot 540) \mod 660 = 60 < 420 .
\]

Conclusion: the residue $a_4$ modulo $m_4$ of the number $\hat{A}_4$ is possibly a corrupted residue $\overline{a}_4$.

\[
\hat{A}_{5PNS} = \left( \sum_{i=1}^{n} a_i \cdot B_{j5} \right) \mod M_5 .
\]

Since $M_5 = M = 420$ , the residue $\overline{a}_5$ of the check module $m_k = m_5$ will be always among the totality of possibly corrupted residues $\overline{a}_i$ of RC number.

Overall conclusion. During data diagnosing of $\hat{A} = (0 \| 0 \| 0 \| 0 \| 0 )$ in NCS, the residues $a_2 = 0$ and $a_3 = 0$ were ensured not being corrupted. The residues to the bases $m_1$, $m_4$ and $m_5$ might be corrupted, i.e. the residues $\overline{a}_1 = 0$, $\overline{a}_4 = 0$ and $\overline{a}_5 = 5$.

\[
B_4 = (0 \| 0 \| 0 \| 1 \| 0 ) = 2640 , \overline{m}_4 = 4 ;
\]

\[
B_5 = (0 \| 0 \| 0 \| 0 \| 1 ) = 2520 , \overline{m}_5 = 6 .
\]

I. Data verification of $A_{RC} = (0 \| 0 \| 0 \| 0 \| 5 )$. According to the control procedure [1] the value will be defined as:

\[
A_{PNS} = \left( \sum_{i=1}^{n+1} a_i \cdot B_i \right) \mod M_0 =
(0 \cdot 1540 + 0 \cdot 3465 + 0 \cdot 3696 + 0 \cdot 2640 +
5 \cdot 2520) \mod 4620 =
12600 \mod 4620 = 3360 > 420 .
\]

Thus, in the process of data verification it was evaluated, that $A_{RC} = 3360 > M = 420 .
$

In this case, with the possibility of only single errors appearing, conclusion is made that the number in question $\hat{A}_{3360} = (0 \| 0 \| 0 \| 0 \| 5 )$ is incorrect ($3360 > M = 420$).

In order to correct the number $\hat{A}_{3360} = (0 \| 0 \| 0 \| 0 \| 5 )$ data is required to be verified first, i.e. corrupted residue $\hat{a}$, has to be detected. Once done, the true value of the residue $a$ modulo $m$ needs to be defined, whereupon the corrupted residue $\hat{a}$ should be corrected.

II. Data diagnosing of $\hat{A}_{3360} = (0 \| 0 \| 0 \| 0 \| 5 )$. According to the mapping method [1, 2], possible projections $\hat{A}_j$ of the number $\hat{A}_{3360} = (0 \| 0 \| 0 \| 0 \| 5 )$ are:

\[
\hat{A}_1 = (0 \| 0 \| 0 \| 0 \| 5 ) , \hat{A}_2 = (0 \| 0 \| 0 \| 0 \| 5 ) ,
\hat{A}_3 = (0 \| 0 \| 0 \| 5 ) , \hat{A}_4 = (0 \| 0 \| 0 \| 0 )
\]

and $\hat{A}_5 = (0 \| 0 \| 0 \| 0 )$.

Computational formula for the values $\hat{A}_{jPNS}$ of PNS number projections is written as [1]:

\[
\hat{A}_{jPNS} = \left( \sum_{i=1; j=a_i, a_i+1} a_i \cdot B_{ji} \right) \mod M_j . \quad (11)
\]

According to the expression (11) we’ll compute all the values of $\hat{A}_{jPNS}$. Once done, we will make $(n+1)$ comparison of the $\hat{A}_{jPNS}$ numbers to the number $M = M_0 / m_{n+1}$. If there are any numbers not being contained in the informational numeric interval $[0, M)$, which contains $k$ correct numbers (i.e. $\hat{A}_k \geq M$), among $\hat{A}_j$ projections, then conclusion is made that these $k$ residues of the number $A$ are not corrupted. Only the residues among the rest $[(n+1) - k]$ number $\hat{A}_{RC}$ residues can be false.
In this case it’s required to correct the residues \( \tilde{\alpha}_1 \), \( \tilde{\alpha}_4 \) and \( \tilde{\alpha}_5 \).

III. Correcting data errors \( \tilde{\lambda}_{3360} = (0 \| 0 \| 0 \| 0 \| 5) \). According to the acquainted [1] expression:

\[
a_i = \left( \tilde{\alpha}_i + \frac{m_i \cdot (1 + r \cdot m_{n+1})}{m_{n+1} \cdot B_i} - \frac{\tilde{\lambda}}{B_i} \right) \mod m_i = \left( 0 + \frac{3 \cdot (1 + r \cdot 11)}{11} \right) \mod 3 = 1;
\]

\[
a_4 = \left( \tilde{\alpha}_4 + \frac{m_4 \cdot (1 + r \cdot m_{n+1})}{m_{n+1} \cdot B_4} - \frac{\tilde{\lambda}}{B_4} \right) \mod m_4 = \left( 10 + \frac{7 \cdot 12}{11} \right) \mod 7 = 0;
\]

\[
a_5 = \left( \tilde{\alpha}_5 + \frac{m_5 \cdot (1 + r \cdot m_{n+1})}{m_{n+1} \cdot B_5} - \frac{\tilde{\lambda}}{B_5} \right) \mod m_5 = \left( 5 + \frac{11 \cdot 11}{11} \right) \mod 11 = 0.
\]

With accordance to the computed residues \( a_1 = 1 \), \( a_4 = 0 \) and \( a_5 = 0 \) we are correcting (recovering) the corrupted number \( \tilde{\lambda}_{3360} = (0 \| 0 \| 0 \| 0 \| 5) \), i.e. the corrected number becomes \( \tilde{\lambda}_{cor} = (1 \| 0 \| 0 \| 0 \| 5) \).

To validate corrected data, as according to the acquainted [1] expression, we’ll define the value of the number \( \tilde{\lambda}_{cor,PNS} = (1 \| 0 \| 0 \| 0 \| 5) \) in the following way:

\[
\tilde{\lambda}_{cor,PNS} = \sum_{i=1}^{5} a_i \cdot B_i \mod M_0 = (a_1 \cdot B_1 + a_2 \cdot B_2 + a_3 \cdot B_3 + a_4 \cdot B_4 + a_5 \cdot B_5) \mod M_0 = (1 \cdot 11540 + 1 \cdot 3360 + 0 \cdot 2640 + 5 \cdot 2520) \mod 4620 = 14140(\mod 4620) = 280.
\]

Thus \( 280 < M = 420 \), the number \( \tilde{\lambda}_{3360} = (1 \| 0 \| 0 \| 0 \| 5) \) is correct.

In order to validate correctness of the number \( \tilde{\lambda}_{3360} \) we’ll make a computation and comparison of the values to the correct residues \( a_2 = 0 \) and \( a_5 = 0 \). In this case they are

\[
a_2 = \left( 0 + \frac{4 \cdot (1 + 11)}{11} \right) \mod 4 = 0
\]

and

\[
a_5 = \left( 0 + \frac{5 \cdot (1 + 11) - 3360}{11 - 4} \right) \mod 5 = 0.
\]

The resulted computations \( a_2 = 0 \) and \( a_5 = 0 \) of the residues modulo \( m_2 \) and \( m_5 \) in RC verified correctness of the corrupted number \( \tilde{\lambda}_{3360} = (0 \| 0 \| 0 \| 0 \| 5) \). Thus, the original number \( \tilde{\lambda}_{RC} = (0 \| 0 \| 0 \| 0 \| 5) \) is corrupted \( \tilde{\lambda}_{3360} \), wherein the single error \( \Delta \alpha_1 = 1 \) occurred modulo \( m_1 \). This error made the correct number \( A_{280} \) being corrupted \( \tilde{\lambda}_{3360} \).

In order to verify if the correct number \( A_{280} \) is true, subsidiary tests on the process of corruption and correction of the number \( A_{280} \) modulo \( m_1 = 3 \) are required. The amount of possible \( N_{CC} \) incorrect (corrupted) \( \tilde{\lambda}_{RC} \) codewords (if only a single error occurred) for each correct \( A_{RC} \) number are

\[
N_{CC} = \sum_{i=1}^{n+1} m_i - (n + 1).
\]

Test results have shown that corruption of the residue \( a_1 \) modulo \( m_1 = 3 \) of the correct number \( A_{280} \) can produce only two incorrect numbers: \( \tilde{\lambda}_{3360} = (0 \| 0 \| 0 \| 0 \| 5) \) and \( \tilde{\lambda}_{3360} = (1 \| 0 \| 0 \| 0 \| 5) \). This points to the fact that the corrected number \( A_{cor} = A_{280} = (1 \| 0 \| 0 \| 0 \| 5) \) is both correct (is contained in the interval \([0, 420]\)) and true. The trueness of the resulted number \( A_{280} = (1 \| 0 \| 0 \| 0 \| 5) \) is confirmed by the fact that the single error \( \Delta \alpha_1 = 2 \) to the base \( m_1 = 3 \) converts \( \tilde{\lambda} = (A + \Delta \lambda) \mod M_0 = (1 \| 0 \| 0 \| 0 \| 5) \).

Example №2. Assume, the correct number is \( A_{280} = (1 \| 0 \| 0 \| 0 \| 5) \) and assume that \( \Delta \alpha_1 = 1 \). In this case

\[
\tilde{\lambda} = (A + \Delta \lambda) \mod M_0 = (1 \| 0 \| 0 \| 0 \| 5) + (1 \| 0 \| 0 \| 0 \| 5) = (1 \| 1 \| 0 \| 0 \| 5) \mod 3 \| 0 \| 0 \| 0 \| 5) = (2 \| 0 \| 0 \| 0 \| 5).
\]

This RC number is relevant to the number 1820 in PNS, i.e. the number \( \tilde{\lambda}_{1820} \) is incorrect. We’ll correct the number \( \tilde{\lambda}_{1820} \) now.

Data diagnosing should be made ahead of correcting the number \( \tilde{\lambda}_{1820} \). To do this we’ll map projections \( \tilde{\lambda}_i \) of the number \( \tilde{\lambda}_{1820} = (2 \| 0 \| 0 \| 0 \| 5) \) first. Resulted RC code structures are:

\[
\tilde{\lambda}_1 = (0 \| 0 \| 0 \| 0 \| 5), \quad \tilde{\lambda}_2 = (2 \| 0 \| 0 \| 0 \| 5), \quad \tilde{\lambda}_3 = (2 \| 0 \| 0 \| 0 \| 5), \quad \tilde{\lambda}_4 = (2 \| 0 \| 0 \| 0 \| 0).
\]
All the projections of $\tilde{A}_{\text{PNS}}$ are:

- $\tilde{A}_{\text{PNS}} = (5 \cdot 980) \mod 1540 = 280 < 420 = M$;
- $\tilde{A}_{2\text{PNS}} = 1925 \mod 1155 = 770 > 420 = M$;
- $\tilde{A}_{3\text{PNS}} = 4592 \mod 924 = 896 > 420 = M$;
- $\tilde{A}_{4\text{PNS}} = 3140 \mod 660 = 500 > 420 = M$;
- $\tilde{A}_{5\text{PNS}} = 560 \mod 420 = 140 < 420 = M$.

Inasmuch as $\tilde{A}_{2\text{PNS}}, \tilde{A}_{3\text{PNS}}$ and $\tilde{A}_{4\text{PNS}} > 420$, the conclusion is made that the residues $a_2 = 0, a_3 = 0$ and $a_4 = 0$ of the number $\tilde{A} = (2 \| 0 \| 0 \| 0 \| 5)$ are not corrupted. Only the residues $a_1$ and $a_5$ can be corrupted $\tilde{a}_1 = 2$ and $\tilde{a}_5 = 5$.

We obtain that:

$$a_1 = \tilde{a}_1 + \left[ \frac{m_1 \cdot (1 + r \cdot m_2 + 1) - A}{B_1} \right] \mod m_1 =$$

$$\left(2 + \frac{3 \cdot (1 + 11)}{11 - 1} \cdot 1820 \right) \mod 3 =$$

$$\left(2 + [3, 27 - 1, 18] \right) \mod 3 = 4 \mod 3 = 1.$$

Hence, the corrected residue modulo $m_1$ is $a_1 = 1$. In a like manner the residue $a_5 = 5$. Applying the results $a_1$ and $a_5$ the corrected number

$$\tilde{A}_{280} = (2 \| 0 \| 0 \| 0 \| 5)$$

is corrected. As a final result the corrected number is

$$A_{280} = (1 \| 0 \| 0 \| 0 \| 5).$$

Conclusions of research

Contrary to PNS (positional numeral system), arithmetic RC (residue class) codes feature additional correcting properties. Thus, NCS (non-positional code structure) involves both intrinsic and subsidiary information redundancies, that in some cases results in allowing to correct single errors in RC, while MCD is $d_{\text{min}}^{(RC)} = 2$. However, correcting single errors requires performing subsidiary tests of data checking, i.e. time redundancy usage, additionally to information redundancy. Examples of specific implementation of a single error correcting procedures were introduced, that prove reviewed method is possible to be implemented in order to correct data errors in RC.

REFERENCES

1. Akushskii, I. Ya., Yuditskiy, D.L. (1968), Arithmetic in the residual classes, Sov. radio, 440 p.
2. Krasnobayev, V., Kurmetsov, A., Lokotkova, I., and Dyachenko, A. (2019), "The Method of Single Errors Correction in the Residue Class," 2019 3rd International Conference on Advanced Information and Communications Technologies (AICT).
3. Tariq Jamil (2013), Complex Binary Number System. Algorithms and Circuits. India: Springer. 83 p.
4. Ananda Mohan (2016), Residue Number Systems. Birkhäuser Basel. 351 p.
5. Cheryvayok, N. I. "Residue-to-binary conversion for general moduli sets based on approximate Chinese remainder theorem", International Journal of Computer Mathematics. 2017. T. 94, №. 9. С. 1833-1849.
6. Kasianchuk, M., Yakymenko, I., Pazdriy, I. and Zastavnyy, O. (2015), "Algorithms of findings of perfect shape modules of remaining classes system," The Experience of Designing and Application of CAD Systems in Microelectronics, Lviv, pp. 316-318. doi: 10.1109/CADSM.2015.7230866.
7. Krasnobayev, V. A., Koshman, S. A. and Mavrina, M. A. (2014), "A method for increasing the reliability of verification of data represented in a residue number system", Cybernetics and Systems Analysis, vol. 50, Issue 6, pp. 969-976.
8. Krasnobayev, V. A., Yanko, A. S. and Koshman S. A. (2016), "A Method for arithmetic comparison of data represented in a residue number system", Cybernetics and Systems Analysis, vol. 52, Issue 1, pp. 145-150, January.