UDC 681.3
DOI: 10.15587/2706-5448.2021.233656
Article type «Reports on Research Projects»

Vasyl Semerenko,
Oleksandr Voinalovich

THE SIMPLIFICATION OF COMPUTATIONALS IN ERROR CORRECTION CODING

The object of research is the processes of error correction transformation of information in automated systems. The research is aimed at reducing the complexity of decoding cyclic codes by combining modern mathematical models and practical tools. The main prerequisite for the complication of computations in deterministic linear error-correcting codes is the use of the algebraic representation as the main mathematical apparatus for these types of codes. Despite the universalism of the algebraic approach, its main drawback is the impossibility of taking into account the characteristic features of all subclasses of linear codes. In particular, the cyclic property is not taken into account at all for cyclic codes. Taking this property into account, one can go to a fundamentally different mathematical representation of cyclic codes – the theory of linear automata in Galois fields (linear finite-state machine).

For the automaton representation of cyclic codes, it is proved that the problem of syndromic decoding of these codes in the general case is an NP-complete problem. However, if to use the proposed hierarchical approach to problems of complexity, then on its basis it is possible to carry out a more accurate analysis of the growth of computational complexity. Correction of single errors during one time interval (one iteration) of decoding has a linear decoding complexity on the length of the codeword, and error correction during m iterations of permutations of codeword bits has a polynomial complexity. According to three subclasses of cyclic codes, depending on the complexity of their decoding: easy decoding (linear complexity), iteratively decoded (polynomial complexity), complicate decoding (exponential complexity). Practical ways to reduce the complexity of computations are considered: alternate use of probabilistic and deterministic linear codes, simplification of software and hardware implementation by increasing the decoding time, use of interleaving. A method of interleaving is proposed, which makes it possible to simultaneously generate the burst errors and replace them with single errors. The mathematical apparatus of linear automata allows solving together the indicated problems of error correction coding.

Keywords: *cyclic codes, NP-completeness, computational complexity, cyclic permutation, iterative decoding.*

Received date: 29.01.2021
Accepted date: 12.03.2021
Published date: 30.06.2021

How to cite
Semerenko, V., Voinalovich, O. (2021). The simplification of computationals in error correction coding. Technology Audit and Production Reserves, 3 (2 (59)), 24–28. doi: http://doi.org/10.15587/2706-5448.2021.233656

1. Introduction

The history of error correction coding dates back to the 1948 publication of the well-known work [1, 2]. The first decades after that were characterized by the rapid development of a new branch of science, the creation of many currently known codes. But there was no practical implementation of new developments. The costs of implementing encoders and decoders could not be recouped in the then transmission and storage channels. Only thanks to satellite and space communications, where various innovations were quickly introduced, a new field of science and technology could develop [3].

The situation changed rapidly in the early 90s in connection with the emergence of new telecommunication systems and various means of microcircuitry. This made it possible to immediately introduce powerful turbo codes and implement the Gallager codes (current name: (low-density parity-check (LDPC) codes) [4]), which have been waiting in the wings for more than 30 years.

However, despite a number of positive characteristics of the new codes, the complexity of their implementation was also quite high.

Theorists of error correction coding were not interested in the problem of complexity for a long time, the first publications on this topic appeared only in the 70s [5]. Meanwhile, engineers have proposed many different ways to simplify computations, for example, the permutation operation [6].

As a result, a rather motley picture of the combination of theoretical methods and practical methods of simplifying the operations of code conversion has now emerged. Therefore, the task of analyzing the problem of computational complexity in modern error correction coding is timely and urgent.

Thus, *the object of research* is the processes of error correction transformation of information in automated systems. *The aim of research* is to reduce the complexity of decoding cyclic codes by combining mathematical models and practical tools.

2. Methods of research

Further, the problem of complexity was investigated by other mathematicians, and they all agreed that the main problems of error correction coding belong to NP-hard (NP-complete) problems [7]. Many experts interpreted these results as such, confirming the impossibility of a simpler decoding algorithm, which would be much simpler than brute force search over all codewords or all syndromes.

Of course, the wider the class of codes is investigated, the more arguments can be made for the growth of computational complexity. But we work mainly with one type of codes, and it is this type of codes that needs to be investigated and search for polynomial or at least subexponential decoding algorithms for it.

Considering the wide scope of application of cyclic codes, let's restrict ourselves only to this class of codes. To represent cyclic codes as a subclass of linear codes, algebraic or matrix methods are traditionally used. And without a deep mathematical foundation, it is clear that solving systems of algebraic equations quickly turns the non-deterministic algebraic Berlekamp-Massey algorithm into an exponential in complexity.

Therefore, for cyclic codes let's use the most suitable mathematical apparatus for them – the theory of linear automaton, that is, linear finite-state machine (LFSM) [8].

DEFINITION. LFSM is a finite automaton of linear type, over the Galois field is described by the transition function:

$$S(t+1)=A\cdot S(t)+B\cdot U(t),\ GF(q),$$

and the output function:

$$Y(t)=C\cdot S(t)+D\cdot U(t),\ GF(q),$$

where t – discrete time; A, B, C, D – characteristic matrices; $S(t)$ – state word; $U(t)$ – input word; $Y(t)$ – output word.

The automaton representation takes into account the main property of cyclic codes – the property of cyclicity. This makes it possible to consider the automaton-analytical and automaton-graph models of a cyclic code, and develop, on their basis, fundamentally new coding and decoding algorithms of predominantly polynomial complexity.

As proved in [8], a cyclic (n,k)-code is able to correct all random errors of multiplicity or less if:

1) its automaton-graph model has $\tau_{\min}$ levels;

2) at the τ-th level, the total number N_τ of vertices of all zero cycles (ZC) is equal to:

$$N_\tau=\frac{n!}{\tau!(n-\tau)!},\ \tau=1\ldots\tau_{\min}.$$

Let's estimate the complexity of the implementation and the decoding procedure for cyclic codes from formal positions – NP-complete problems. As is known, the NP (*nondeterministic polynomial*) class consists of problems that can be quickly verified (within polynomial time), and the P (*polynomial*) class consists of problems that can be quickly completed [9].

In the case of an automaton representation of cyclic (n,k)-codes, the decoding problem in the general case can be interpreted as constructing a path of length n along a directed graph from another given vertex to another vertex. The number of computations during decoding increases exponentially with the increase in the number of vertices in the graph. The decoding complexity condition is satisfied.

To check the correctness of decoding, it is enough to replace the error bits Z_{err} in the n-bit word and calculate the new value of the error syndrome S_{err}. The parameter S_{err} is the next state of the LFSM. The zero value of the syndrome means the correctness of the decoding operation, otherwise, the presence of errors in the received word Z_{err}. The algorithm for calculating the syndrome is deterministic and always takes n clock cycles of the decoder. The simple check condition is also met.

Although the automaton methods of decoding are generally NP-complete problems, for many subclasses of cyclic codes decoding algorithms have polynomial complexity.

For example, the most common cyclic codes – CRC codes (Cyclic Redundancy Code) – can correct single errors $(\tau=1)$ or detect all double and odd errors for any code length n [10]. Finding such errors will take no more than n clock cycles of the decoder. Therefore, for CRC codes, the decoding complexity will always be linear $O(n)$, and the codes themselves can be called easily decoded. Further, let's consider other types of codes with simple decoding.

As in many other areas, in practice, the problem of computational complexity can be solved by breaking the overall complex problem into a number of simple subtasks. In error correction coding, it is advisable to analyze the following subproblems:

- combination of several error correction codes;
- use of iterative procedures;
- use of permutation.

Let's consider critically each of these subtasks and the degree to which it performs the function of simplifying computations for various types of codes.

3. Research results and discussion

3.1. Combination of several error correction codes. According to the author of [1, 2] the main condition for obtaining a code with a high correction ability is the maximum possible length of this code. However, the complexity of the implementation of such a code will grow exponentially.

The way out of this deadlock was the idea of concatenation of several (more often two) deterministic codes, proposed by the author of [11]. According to this proposal, the codewords of the code of the previous coding stage can be considered as the original data for the next coding stage, and the decoding is carried out in the reverse order. In the language of mathematics, the direct (Kronecker) product of the (n_1,k_1)-code Ω_1 and the (n_2,k_2)-code Ω_2 gives the (n_1n_2,k_1k_2)-code Ω. In a concatenated code, an increase in the correction ability occurs when the code rate decreases and the complexity of the source codes Ω_1 and Ω_2 remains unchanged.

A completely different situation arises when combining deterministic and probabilistic codes. The purpose of this combination is to improve the performance of new code. For example, VoIP Internet telephony systems use data formats with a length of 40 to 352 bits [12]. For such code lengths, the performance of turbo codes decreases sharply, that is, increases with each subsequent error. On the other hand, in such data formats, the deterministic CRC quickly corrects individual errors left behind by the probabilistic code. Similarly, digital television combines LDPC codes with Bose-Caudhuri-Hochquenghem (BCH) codes.

3.2. Iterative implementation of the decoding procedure. According to the theory of algorithms [13], each computational algorithm can be evaluated according to two criteria:

1) time complexity;
2) space complexity.

In terms of error correction coding, space complexity is estimated by the cost of hardware or software implementation of the encoding and decoding operations.

The first codes did not require a lot of time or hardware. Relatively speaking, for the first codes, the implementation of these operations provided for a single execution of the encoding and decoding algorithms during one time interval. However, decoding any code in one pass is extremely expensive.

The theory of algorithms suggested a way to solve this problem: a decrease in space complexity is possible at the expense of an increase in time complexity. Once the probabilistic algorithms were implemented in a rather iterative mode, it was then possible to obtain practical encoders/decoders for LDPC codes and turbo codes. space complexity was exchanged for increased encoding/decoding times.

But to this time, iterative decoding is associated only with probabilistic codes. In [14], the use of iterative decoding is substantiated for deterministic, in particular, cyclic codes. For this approach, it is possible to use non-algebraic decoding methods in Galois fields. As the author of [15] showed, for any integer all cyclic codes are invariant under permutations of symbols of the form:

$$i \to (q^{\upsilon} i) \bmod n,\ GF(q^m).$$

In other words, $g(x)$ if the generator polynomial $g(x)$ of the cyclic code divides the code polynomial $f(x)$, then it will also divide the polynomial $f(x^q)$, the symbols of which are rearranged in accordance with the rule $i \to q^i$. Let's call such permutations as cyclic. The most simple cyclical permutation of a codeword Z is that the odd bits of the word are written first, and then the even bits (you can start with even bits too).

As a result, let's obtain a new codeword. The cyclic permutation can be performed sequentially several iterations until the original word Z is obtained (Fig. 1 shows the first three word Z permutations).

In other words, if the generator polynomial $g(x)$ of the cyclic code divides the code polynomial $f(x)$, then it will also divide the polynomial $f(x^q)$, the symbols of which are rearranged in accordance with the rule $i \to q^i$.

It is important that with the help of such permutations, the error correction in the received codeword is directly implemented.

Iteration 1

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

Iteration 2

1	3	5	7	9	11	13	15	17	2	4	6	8	10	12	14	16

Iteration 3

1	5	9	13	17	4	8	12	16	3	7	11	15	2	6	10	14

Fig. 1. Cyclic permutations of a 17-bit codeword

Let there be noise in the communication channel, which caused a set of errors in the codeword Z_{err} within the correction capacity of a specific (n,k)-code. It is possible to introduce the concept of the checking windows X as a continuous cyclic sequence of r bits of an n-bit word $(r = n - k)$ (Fig. 2).

$$X^{(v)} = z_v \dots z_w\,,\quad z_i \in Z,\ w = v + r - 1) \bmod n,\quad i = 1 \div n.$$

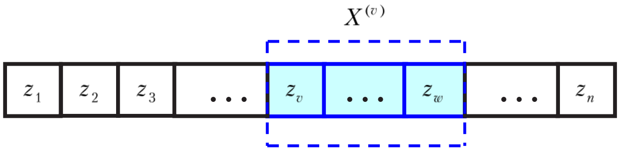


Fig. 2. Codeword Z_{err} checking window $X^{(i)}$

Such a window $X^{(i)}$ can be cyclically moved along the entire word Z_{err}. The essence of the method for correcting multiple errors consists in calculating a new error syndrome S_{err} at each iteration and forming such a variant of the permutation when all the error bits of the codeword Z_{err} fall into the checking window $X^{(i)}$. A sign of correct decoding is obtaining a zero value of the error syndrome S_{err} at one of the iterations.

Let's estimate the complexity of decoding using cyclic permutations. First, let all τ erroneous bits in the word Z_{err} be located in different checking windows $X^{(i)}$. If it took h iterations of cyclic permutations and n clock cycles of the decoder at each iteration, then the complexity of decoding the (n,k)-code will be quadratic $O(nh)$ $(h \leq n)$, and the cyclic codes themselves can be called permutable decoded. Such codes include, for example, Golay codes.

The most difficult variant of decoding will be in the case when one iteration fails to place all errors in single checking window. In this case, the LFSM theory allows to use another variant of decoding using a tree-like directed graph $G(V,E)$ of a cyclic code with a set of vertices V and a set of arcs E. Then the decoding result will be successful if it is possible to construct a path of length n along the graph $G(V,E)$ from the vertex corresponding to the syndrome S_{err} to the root vertex. If the known methods for constructing paths in a graph can only offer a complete enumeration of options, then the proposed method in this situation gives recommendations for accelerated path construction based on taking into account the structure of the graph $G(V,E)$.

The complexity of such a search procedure will decrease n times and it can be estimated by the formula:

$$O\left(\frac{[V]}{n}, \frac{[E]}{n}\right).$$

In this case, the complexity of decoding the (n,k)-code will be subexponential, and the cyclic codes themselves can be called as complicated-decoded.

The considered decoding method makes it possible to correct errors within the correction capacity of the cyclic code. Errors beyond the correction capacity can be corrected by using two cyclic codes with different generator polynomials. The first cyclic code is used to search for possible configurations of errors in a word, and the second code is used to confirm the correctness of the selected error configurations [8].

3.3. Use of permutation in error correction coding. The complexity of the calculations depends on the type of errors in the data transmission channels.

Distinguish between inverse errors (changing the correct values to others from a given alphabet of values) and erasures (errors in which the values are unknown, but their location is known). Inverse errors and erasures can be either random (i. e., spaced across the entire length of the word Z_{err}) or burst error (concentrated in a limited area of the word Z_{err}).

In error-correcting coding, there are different interpretations of the permutation operation, the most common are:

- conversion of burst error into single random errors;
- change the order of the positions of code word when it is transmitted over the communication channel (rearrangement of bits);
- special methods of change the positions of code word (interleaving), which are used in turbo-codes and LDPC-codes.

Burst errors must first be converted into statistically random single errors (that is, execute permutation), and then the decoder must perform the actual decoding operation – this is the essence of the Shannon error correction coding strategy. For probabilistic codes, permutation is inalienable operation at encoding and decoding.

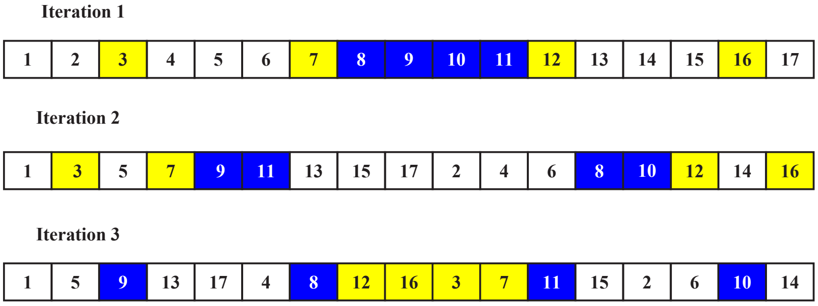
In deterministic codes, permutation is optional, but can also be useful when used correctly [6].

It is believed that the burst error are more difficult to decode (of course, if to use algebraic decoding methods such as Berlekamp-Massey) [4], although in most cases this is a false thesis. For example, a cyclic (n,k)-code using non-algebraic methods can display burst error of length up to $(n-k)$ bits in one codeword Z_{err} and correct burst error of length to $m/2$ bits $(m=n-k)$ [7]. Therefore, in many cases it is advisable to form burst error from single errors, to speed up the process of detecting and correcting erroneous bits in a word Z_{err}.

The mathematical analysis of cyclic permutation shows its interesting and very important properties. First, there is move word positions simultaneously in two opposite directions: n bits of the word gradually merge into burst of length n (Fig. 3). The secondly, at the same time as changing the order of the positions of code word decoder correct errors in this word.

If several codewords are involved in traditional permutation, then the proposed permutation occurs only within one codeword (Table 1)

Another significant difference is that permutation is necessary only on the receiver side. The transmitter transmits the codeword Z to the channel without transformations, significantly saving time. The advantage of (LFSM) is the ability to simultaneously permutation and correct errors.



Iteration 1

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

Iteration 2

1	3	5	7	9	11	13	15	17	2	4	6	8	10	12	14	16

Iteration 3

1	5	9	13	17	4	8	12	16	3	7	11	15	2	6	10	14

Fig. 3. Three-iteration formation of a burst from the positions of the codeword (yellow color) and splitting the second burst from the positions of the codeword for separate positions (blue color)

Table 1

Distinctive features of the two types of permutation

Main characteristics of permutation	Matrix block permutation	Cyclic permutation
Type of transformation	Error burst error→random errors	Error burst error→random errors, random errors→burst error
Code operations	Coding, decoding	Decoding
Object of permutation	Set of codewords	Single codeword

4. Conclusions

The paper shows that traditionally complexity studies are performed for linear codes that combine a large number of different error correction codes, often with mutually opposite characteristics. Of practical interest is a comparative analysis of the complexity of decoding a smaller set of codes with similar characteristics, in particular, a class of cyclic codes. As expected, it confirms the conclusion previously obtained by other researchers that the problem of syndromic decoding of cyclic codes is an NP-complete. In connection with this statement, it is possible to mention the fact that «it is practically impossible to correct more than six errors in one codeword of the Reed-Solomon code» [16]. On the other hand, it is generally known that the main advantage of cyclic codes is the simple implementation of their encoders and decoders. This mathematical paradox can be explained only based on a hierarchical approach to complexity problems, taking into account various factors (type and multiplisity of errors in the channel, suitability of the mathematical apparatus for specific codes, etc.). As a result, different subclasses of cyclic codes have differ degrees of spatial and temporal complexity. Let's believe that in order to reduce the gap between theory and practice, it is advisable to use various practical ways to simplify computations: concatenated codes, iterative decoding, and permutation. So far, not all reserves of effective use of error correction codes have been used.

It should be emphasized that the basic mathematical apparatus of error correction coding have key importance in reducing the complexity of computations, in this case, the theory of linear automaton. Only the transition to non-algebraic representations of cyclic codes allows the use of new approaches in iterative decoding and permutation.

The results of the study are also of interest for other error correction codes.

References

1. Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal, 27 (3),* 379–423. doi: http://doi.org/10.1002/j.1538-7305.1948.tb01338.x
2. Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal, 27 (4),* 623–656. doi: http://doi.org/10.1002/j.1538-7305.1948.tb01338.x
3. Costello, D. J., Hagenauer, J., Imai, H., Wicker, S. B. (1998). Applications of error-control coding. *IEEE Transactions on Information Theory, 44 (6),* 2531–2560. doi: http://doi.org/10.1109/18.720548
4. Morelos-Zaragosa, R. H. (2002). *The Art of Error Correcting Coding.* Jon Wiley & Sons, 278. doi: http://doi.org/10.1002/0470847824
5. Berlekamp, E., McEliece, R., van Tilborg, H. (1978). On the inherent intractability of certain coding problems (Corresp.). *IEEE Transactions on Information Theory, 24 (3),* 384–386. doi: http://doi.org/10.1109/tit.1978.1055873
6. Melentev, O. G. (2007). *Teoreticheskie aspektyi peredachi dannyih po kanalam s gruppiruyuschimisya oshibkami.* Moscow: Goryachaya liniya–Telekom, 232.
7. Vardy, A. (1997). The intractability of computing the minimum distance of a code. *IEEE Transactions on Information Theory, 43 (6),* 1757–1766. doi: http://doi.org/10.1109/18.641542
8. Semerenko, V. P. (2015). Estimation of the correcting capability of cyclic codes based on their automation models. *Eastern-European Journal of Enterprise Technologies, 2 (9 (74)),* 16–24. doi: http://doi.org/10.15587/1729-4061.2015.39947
9. Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stain, C. (2014). *Introduction to Algorithms.* London: The MIT Press, 1183.
10. Semerenko, V. (2016). The theory of parallel CRC codes based on automaton models. *Eastern-European Journal of Enterprise Technologies, 6 (9 (84)),* 45–55. doi: http://doi.org/10.15587/1729-4061.2016.85603
11. Forney, G. D. (1967). *Concatenated Codes.* Cambridge: MIT Press.
12. Wei, Y., Jiang, M., Xia, B., Chen, W., Yang, Y. (2013). A CRC-Aided Hybrid Decoding Algorithm for Turbo Codes. *IEEE Wireless Communications Letters, 2 (5),* 471–474. doi: http://doi.org/10.1109/wcl.2013.052813.130259
13. Aho, A. V., Hopcroft, J. T., Ullman, J. D. (1976). *The Design and Analysis of computer Algorithms.* Addison-Wesley Publishing Company Reading.
14. Semerenko, V. (2018). Iterative hard-decision decoding of combined cyclic codes. *Eastern-European Journal of Enterprise Technologies, 1 (9 (91)),* 61–72. doi: http://doi.org/10.15587/1729-4061.2018.123207
15. Prange, E. (1962). The use of information sets in decoding cyclic codes. *IEEE Transactions on Information Theory, 8 (5),* 5–9. doi: http://doi.org/10.1109/tit.1962.1057777
16. Clark, G. C. Jr., Cain, J. B. (1981). *Error-Correction for Digital Communications.* New York and London: Plenum Press, 422. doi: http://doi.org/10.1007/978-1-4899-2174-1

Vasyl Semerenko, *PhD, Associate Professor, Department of Computer Technique, Vinnytsia National Technical University, Vinnytsia, Ukraine, e-mail: VPSemerenko@ukr.net, ORCID: https://orcid.org/0000-0001-8809-1848*

Oleksandr Voinalovich, *Postgraduate Student, Department of Computer Technique, Vinnytsia National Technical University, Vinnytsia, Ukraine, e-mail: sashavoinalovich@gmail.com, ORCID: https://orcid.org/0000-0003-3695-8918*