Tenth International Symposium on Communication Theory and Application (ISCTA'09)

- **Home**
- **Copyright**
- **Conference Information**
- **Registration**
- **Daily Schedule**

Monday 13th July
- SESSION 1A: CODING I
- SESSION 1B: CODING II
- SESSION 2A: COMMS SYSTEMS & NETWORKS
- SESSION 2B: COMMS SYSTEMS & NETS II

Tuesday 14th July
- SESSION 3A: CODING III
- SESSION 3B: SOURCE CODING & SECURITY I
- SESSION 4A: CODING IV
- SESSION 4B: SIGNAL PROCESSING I
- SESSION 4C: SOURCE CODING & SECURITY II
- SESSION 4D: COMMS SYSTEMS & NETS III

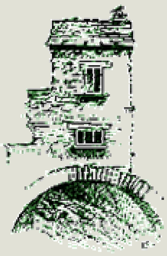
Wednesday 15th July
- SESSION 5: LDPC I
- SESSION 6: COMMS SYSTEMS & NETS IV

Thursday 16th July
- SESSION 7: CODING V
- SESSION 8: LDPC II
- SESSION 9: COMMS SYSTEMS & NETS V
- SESSION 10: COMMS SYSTEMS & NETS VI

Friday 17th July
- SESSION 11: CODING VI
- SESSION 12: COMMS SYSTEMS & NETS VII

Monday 13th July
- POSTER SESSION I

Tuesday 14th July
- POSTER SESSION II



ISCTA '09

13th - 17th July

St Martin's College, Ambleside

United Kingdom

Non-binary multithreshold decoders of symbolic self-orthogonal codes for *q*-ary symmetric channels

V.V. Zolotarev, G.V. Ovechkin
Space Research Institute RAS, Moscow, The Ryazan State Radioengineering University, Ryazan
zolotasd@yandex.ru, g_ovechkin@mail.ru

Abstract – *Symbolic q-ary multithreshold decoding (qMTD) of q-ary self-orthogonal codes is submitted. It's shown the BER performance of qMTD is close to the results provided by optimum total search methods, which are not realizable for non-binary codes in general. New decoders are compared with different decoders of Reed-Solomon codes. The performance provided by non-binary MTD in some cases is unattainable for arbitrary long Reed-Solomon codes with classical decoders. Concatenation of q-ary self-orthogonal code with non-binary single check on module q codes is described. The complexity of qMTD is discussed also.*

Keywords: iterative decoding, multithreshold decoding, symbolic MTD, *q*-ary codes.

Introduction

In many publications [1, 2, 3] it's shown the performance of multithreshold decoders (MTD) over binary Gaussian channels in many cases appears conterminous with the performance of an optimum decoder or close to it even at high noise level. It is well known, that in case of binary codes process of decoding is convenient for considering as finding the global minimum of a multi-variate objective function. It is defined by that the decision of an optimum decoder (OD) is always the code word nearest to the received message in the chosen metrics. Application of MTD type decoders allows at minimal, linear implementation complexity (in terms of the number of operations) to reach the optimum decision even at rather high noise level. And though MTD is not OD the correct choice of code and parameters of MTD algorithm usually allows to achieve almost always the best (optimum!) decision at high noise level. It allows to build software MTD decoders which ones are approximately in hundreds times faster, than other decoding algorithms comparable by performance [3]. Hardware MTD versions implemented on simple serial Xilinx or Altera FPGAs show practically unlimited throughput even in case of data transmission through high-speed channels with large noise level [1].

In some systems it is convenient to work with data having a byte structure. Until recently there were no effective and simultaneously enough simple decoding methods for non-binary (symbolic) codes, except decoders for Reed-Solomon (RS) codes. However short RS codes of length up to n=255 symbols do not provide levels of reliability necessary now. Decoders for long RS codes appear too complex and their essential simplification is rather problematic. There real correcting possibilities are very restricted also.

Recently many experts began to develop decoders for q-ary low-density parity-check (qLDPC) codes [4–6]. The given methods, certainly, possess very high correcting abilities. However complexity of their implementation, especially at large alphabet size q, appears too high for practical application.

Below generalization of MTD on q-ary symmetric channels (qSC) [7–12] is offered. Value of this method consists that majority algorithms have only linear computational complexity, as usually optimum methods are characterized by exponential complexity. Therefore application of q-ary MTD, designated further as qMTD, is represented especially useful.

It is even more essential, that in the case of large q it is absolutely impossible to create effective truly optimum decoders as their complexity may be proportional q^k, where k is the length of an information part of the code. It also defines importance of qMTD application, as opportunities of Reed-Solomon (RS) codes having very wide scope of application are limited by short codes.

Below it will be shown, that qMTD algorithm can provide at rather high noise level the decoder error probability in some cases absolutely inaccessible even for long RS codes. The principle of global functional optimization is generalized on easy implemented decoders of majority decoded non-binary codes. Thus the complexity of such algorithm will be the extremely insignificant, linearly growing with the code length, i.e. theoretically minimal possible.

1. Main properties of *q*MTD

Now consider more formalized description of qMTD decoding algorithm for q-ary codes [3, 12]. For qSC channel with the alphabet size $q > 2$ and the symbol error probability $P_0 > 0$ the optimum decoder decision will be such, probably, unique code word among q^{nR} possible ones which differs from the received message in the minimal number of code symbols. Here n is the code length in symbols and R is the code rate.

Note that later following notations are used: $\mathbf{X}$ – matrix, X – vector, x – variable.

Take a linear q-ary systematic code, which parity-check matrix $\mathbf{H}$ has the same structure as in binary case, except that instead of unit submatrix $\mathbf{I}$ will be placed $-\mathbf{I}$: $\mathbf{H} = [\mathbf{C} : -\mathbf{I}]$.

Let this matrix $\mathbf{H}$ corresponds to q-ary self-orthogonal systematic block or convolutional code [3, 14]. Since parity-check and generating matrixes of the code contain only 0, 1 and –1 then for coding and decoding only addition and subtraction operations are needed. Thus, for coding and decoding non-binary field is not required. For example, all operations of addition and subtraction will be made in some integer group on module q. It simplifies additionally all procedures of coding and subsequent decoding.

After transmission of a code word A of length n with k information symbols over qSC the decoder receives vector Q=A+E, where E is an error vector.

Let as well as in binary case [1, 3], to represent each vector X of length n in the form of pair vectors X_I, X_V of length k and $(n - k)$ accordingly.

Define D as a q-ary vector of length k, equal $D = A_I - Q_I$, where A_I is an information part of the transmitted code word $A = [A_I, A_V]$ and Q_I is an information part of the received message $Q = [Q_I, Q_V]$. Later D is named as differential vector.

Then the following lemma is truth.

Lemma.

$$[D, \mathbf{H}[Q_I + D, Q_V]] = A - Q. \quad (1)$$

The proof. Due to linearity of the code the chain of equalities is fair

$$S = \mathbf{H}[Q_I + D, Q_V] = \mathbf{H}[Q_I + D, Q_V + A_V - A_V] = \mathbf{H}A + \mathbf{H}[0_I, Q_V - A_V].$$

Here 0_I is a zero vector of length k.

Considering, that for a q-ary systematic code $\mathbf{H}[0_I, X_V] = -X_V$, we receive, that $S = A_V - Q_V$. And as $D = A_I - Q_I$ we get $[D, S] = A - Q$.

The lemma is proved.

It establishes simple useful conformity between any code word and the received message, similar to binary case [3, 12]. Actually it approves, that for considered codes the syndrome vector is a difference on check symbols between the received message Q and a codeword with an arbitrary information part A_I. Such interpretation of the syndrome vector was considered from the various points of view in [1, 3, 13].

The lemma allows to prove the main qMTD algorithm property which is described below.

Let due to transmission of a code word A over qSC the qMTD decoder receives vector Q = A + E. Similarly to binary case, the differential q-ary vector D before decoding is zeros vector.

Let further qMTD work in such a way, that after usual calculation the syndrome vector $S = \mathbf{H}Q$ for the received message procedure of decoding consists in the following steps.

1. For any q-ary decoding information symbol i_j the number of two most often meeting values is counted up among all J orthogonal checks concerning the symbol i_j, and also the symbol d_j in the differential vector D. Let values of these two checks are equal h_0 and h_1 ($0 \le h_0, h_1 \le q$), and their quantities are equal m_0 and m_1 accordingly, and $m_0 \ge m_1$.

This procedure is similar to calculation of the sum of checks at the threshold element in the usual binary threshold decoder [15].

2. If $m_0 - m_1 \le T$, where T is a non-negative integer, any new i_m, $m \ne j$, is selected and further transition to step 1 is carried out. It is also analogue of comparison procedure at the threshold element of binary threshold decoder [3, 15].

3. If $m_0 - m_1 > T$ then the estimation of an error equal h_0 is subtracted from i_j, d_j and all J checks for i_j. Then any new i_m, $m \ne j$, is selected and transition to step 1 is carried out.

This last step of the decoding cycle for the next symbol is simply changing of the decoding symbol i_j and correction through a feedback all syndrome symbols, being checks of the decoding symbol, and corresponding symbol in the differential vector D.

Such attempts of decoding in accordance with steps 1...3 can be repeated for each information symbol of the received message, for example, 3, 10 and more times.

At implementation of qMTD algorithm, as well as in binary case, it is convenient to choose all information symbols sequentially and to stop decoding after the fixed number of decoding iterations are executed or if at the next iteration none symbols has been changed.

For the described qMTD algorithm the following theorem is truth.

The theorem. The main theorem of non-binary multithreshold decoding.

Let the decoder realizes algorithm qMTD for a q-ary self-orthogonal code. Then at each change of decoded symbols there is transition to more plausible decision in comparison to the previous decision of the decoder.

Preliminary discussion.

It's known that for a qSC channel among two code words closer to the received message and, hence, more plausible will be that one of them which differs from the received message in less number of symbols. Therefore the proof of growth of qMTD decisions' plausibility at each changing of the decoded symbols is simply a proof, that the number of new codeword's symbols equal symbols of the received message, has increased, i.e. Hamming distance between them has decreased. For non-binary symbols this distance just corresponds to amount of differences in two vectors of equal length.

So, according to properties of a syndrome and differential vectors in accordance with the lemma for *q*MTD the distance between the received vector and the current decision of *q*MTD is equal to number of nonzero symbols in the syndrome and differential vectors. Means, for this distance reduction that correspond to growth of the decoder decisions' plausibility, it is necessary to find other codeword for which the total number of zero symbols of the syndrome and differential vectors will increase. We shall remind that, as well as in binary case, code words are considered which information symbols are equal to an information symbols in the information register of the decoder.

The proof.

Let the *q*MTD decoder contains vectors A_{1I}, $D_1 = A_{1I} - Q_I$ and $S_1 = \mathbf{H}\,[Q_I + D_1, Q_V]$,

Here $A_1 = [A_{1I}, A_{1V}]$ is any codeword and Q is the received message.

We show, that at change the next decoded symbol i_j in the current information vector-decision A_{1I} of the decoder such new vector A_{2I} turns out, that Hamming distance between the received vector Q and the codeword $A_2 = [A_{2I}, A_{2V}]$ is less, than for the previous decision A_1 of the decoder, i.e $|A_1 - Q| > |A_2 - Q|$.

Really, if some symbol i_j was changed, there was a unique value h_0, $h_0 \neq 0$, among values of the checks for the symbol i_j which met strictly most often, m_0 times, and all other values are met no more then m_1 times, $m_0 > m_1$. We note, that if $h_0 = 0$ the decoded symbol does not change.

In this case at change i_j, d_j and all J checks in the syndrome register, i.e. after subtraction from them the value h_0, all m_0 checks (and, maybe, the symbol d_j) which were equal h_0 become equal 0. The amount of zero checks in the syndrome vector (certainly, in view of the symbol d_j) which before the symbol i_j change were equal to zero cannot be more then m_1. But it means that at change i_j the weight of the syndrome vector cannot increase on these positions more than at m_1. Then the total change of weight is equal $m_1 - m_0 < 0$, i.e. total weight of vectors D and S after change of the decoded symbol i_j will decrease.

Also we can notice, that the new syndrome vector after correction differs from old one only in those symbols which are checks for i_j, and the new differential vector differs from old one only in the position d_j at the value h_0, as well as corresponding to the symbol i_j checks. But it means that after change i_j the decoder vectors S_2 and D_2 contain corresponding differences between the received vector and the new decision of the decoder, i.e. the condition (1) is satisfied. But then it appears, that in a new state of the decoder conditions of the lemma are truth again that allows to pass to the next attempt of another symbol i_m, $m \neq j$, correction as a result of which change of a next decoded symbol again guarantees transition to more plausible decision, etc.

The theorem is proved.

We note two most essential moments, describing the offered algorithm. First, as well as in case of binary codes, it is impossible to approve, that improvement of the decision at each decoding iteration will take place until the decision of OD will be reached. Actually both in block and in convolutional codes errors configurations are possible, which are not corrected in *q*MTD but which can be corrected in OD. Therefore the basic way to increase efficiency of *q*MTD consists in codes searching in which such non-corrected errors configurations are rare enough even at a high noise.

Other major moment is that in comparison with the traditional approach to binary majority schemes in *q*MTD for change a decoded symbol it is enough presence only relative strict majority of correct checks as it follows from the condition $m_0 - m_1 > T$. For example, in a self-orthogonal code with $d = 9$ error in a decoded symbol will be corrected even if among 9 its checks (including the symbol d_j of the differential vector) there are only two correct checks and others 7 checks may be erroneous. It is impossible in case of binary codes, but for *q*MTD the given situation is typical. The only condition for this individual example is different values of erroneous checks corresponding to decoded symbol i_j. And for large alphabet size q this condition practically always is filled. These properties of *q*MTD essentially expand opportunities of non-binary multithreshold algorithm at work at high noise, keeping thus rather small complexity of majority procedures in *q*-ary channels.

2. Lower bound for symbol error probability of an optimum decoder for *q*-ary self-orthogonal code

We consider the lower bound calculation for symbol error probability of an optimum decoder for a *q*-ary self-orthogonal code. For this purpose we describe most frequent events at which the error vector will have Hamming distance to the nearest nonzero codeword smaller than its own weight. For linear code it is enough to make a wrong decision even with an optimum decoder. Considering an error vector with such properties, we see that it is necessary to analyze only those symbols of this vector which correspond to the positions of checks concerning the current decoding symbol i_k. We shall write out probabilities of some most simple events which cause errors of OD.

The required events are [3, 10, 12]:

– all check symbols and the decoding symbol i_k are erroneous:

$$P_1(e) = P_0^{J+1}, \quad (2)$$

where $d = J + 1$ is the minimal code distance;

– all check symbols are erroneous, but two of them are identical, and i_k is correct:

$$P_2(e) = \frac{J(J-1)(1-P_0)P_0^J}{2(q-1)} \prod_{i=1}^{J-2}\left(1-\frac{i}{q-1}\right); \quad (3)$$

– one check symbol is received correctly, and the others are erroneous, as well as the symbol i_k:

$$P_3(e) = J(1-P_0)P_0^J. \quad (4)$$

Thus lower bound for symbol error probability is

$$P_{opt} = P_1(e) + P_2(e) + P_3(e). \quad (5)$$

Number of the described events is quite enough to get satisfactory on accuracy estimations of symbol error probability. More full analysis of the various events leading to errors of a non-binary OD, estimations for probabilities of their occurrence, and also first error probability for non-binary threshold decoder are presented in [3, 10, 12].

As qMTD on each step aspires to the decision of OD it is possible to expect, that at some high enough noise level in many cases it will reach the required optimum decision. Thus the complexity of qMTD remains linear function of n, i.e. it is theoretically minimal possible.

3. Performance of *q*MTD over *q*SC

The symbol error rate (SER) performance for decoders of codes with code rate R=1/2 over qSC is shown in fig.1. On the horizontal axis channel SER P_0 is presented and on the vertical axis average SER after decoding is shown. Here curves *3* and *4* correspond to qMTD for codes of length n=4000 and n=32000 one-byte symbols (q=256) accordingly. Dotted line in fig. 1 shows the lower bound P_{opt} for symbol error probability of OD for the first code. It's seen that qMTD can achieve OD performance at rather high noise level. For achievement the optimum decision or close to it, qMTD for q = 256 requires from 5 up to 20 decoding iterations. It completely corresponds to MTD for binary codes [1, 2, 3]. For comparison in fig. 1 the performance of (255, 128) RS code over GF(256) is also shown by curve *1*. As it follows from fig. 1 qMTD provides much better performance than decoder for RS code for symbols of the same size due to greater length of

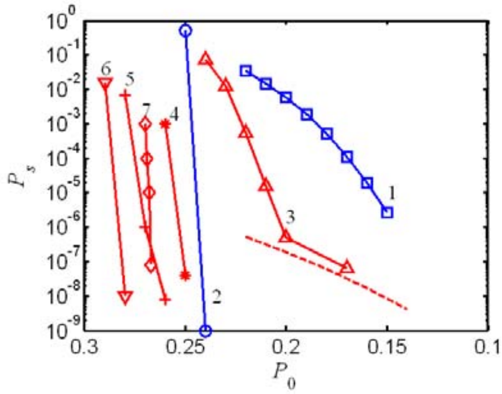


Fig.1. SER performance of rate one-half Reed-Solomon codes and qMTD over qSC

used codes and to good qMTD decisions convergence to the OD decisions. It should be noted, that other decoding algorithms with acceptable complexity besides qMTD which can provide the same performance are unknown now.

Further we describe simulation results for codes with lager alphabet size q. The performance of qMTD for codes with R=1/2, n=32000 and q=2^{16} (two-byte symbols) is presented in fig. 1 by curve *5*. We note that very simple for implementation qMTD for the code of length 32000 symbols appears capable to provide error correcting ability essentially unattainable even for RS code of length 65535 over GF(2^{16}) (curve *2* in fig. 1), a decoder for which will be created never due to its high complexity. Thus qMTD for two-byte symbols practically is not more complex than one-byte one as even usual microprocessors simply and quickly work and with one-byte symbols, and with 2 and even with 4-byte symbols. For example the performance of qMTD for four-byte symbols (q=2^{32}) is shown in fig. 1 by curve *6*.

For communication and data storage systems due to different restrictions high-rate q-ary codes are very useful. The performance of qMTD for codes with R=7/8, n=48000 symbols and q=256 is shown in fig. 2 by curve *3* and performance of RS code with R=7/8 over GF(256) is presented by curve *1*. It's seen that here qMTD outperform RS codes significantly. Similar relation between performance of these error-correction methods remains at using higher code rate R=19/20. For the code rate performance of qMTD with q=256 is shown in fig. 2 by curve *4* and curve *2* presents performance of RS code over GF(256). In this case qMTD is much more effective than RS codes too. From comparison of RS codes with length n = 255, R = 7/8 and R = 19/20 it is clear that the last code is much less effective then the first one and it is much more difficult to provide good efficiency at redundancy reduction. Nevertheless the performance of low redundancy codes with

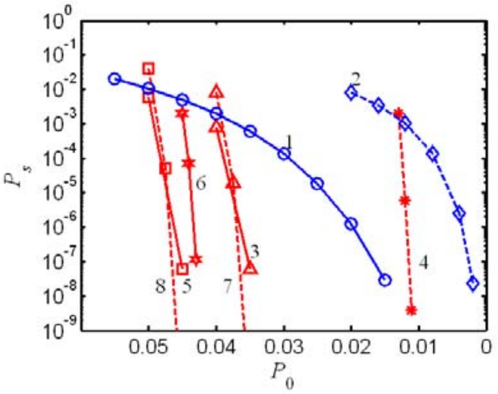


Fig. 2. SER performance of high-rate Reed-Solomon codes and *q*MTD over *q*SC

*q*MTD decoding appears rather high and can essentially increase error correcting ability if the chosen codes have enough large lengths. The performance of *q*MTD for two-byte symbols and R=7/8 is shown in fig. 2 by curve *5*.

It should be noted than for achieving such results with *q*MTD it needs to select of using codes carefully. The main criterion at codes searching is their resistance to error-propagation effect [3, 13]. For illustration of the statement in fig. 1 by curve *7* and in fig. 2 by curve *6* the performance of *q*MTD for codes with q=256, R=1/2 and R=7/8 is presented accordingly. The used codes were selected some more carefully than before. It's seen that performance of *q*MTD for the codes is some better.

Other experimental results for *q*MTD can be found in [9–12].

4. The performance of concatenation with *q*MTD

According to the general principles of the coding theory using of concatenated codes can improve the performance of *q*MTD additionally. Thus the decoding complexity will increase in comparison with base algorithm very slightly.

SER performance of concatenated codes consisting of a q-ary self-orthogonal code and codes with single check on module q is shown in fig. 2 also. Here SER for concatenated codes with an internal q-ary self-orthogonal code of rate $R = 7/8$ and an external code with single check on module q of length $L = 190$ are presented by curve *7* for a code with $q = 256$ (one-byte symbols) and by curve *8* for a code with $q = 2^{16}$ (two-byte symbols). Predictably, application of concatenation in many decimal exponents reduces SER in comparison with base *q*MTD decoder almost without growth of concatenated code redundancy. Thus increasing in complexity of decoder for the concatenated codes is less than 20% in comparison with base *q*MTD algorithm.

The further significant improvement of *q*MTD performance is possible with using convolutional codes, methods of sequential or parallel concatenation, using of codes with allocated branches and by other methods, some of which are described in [3, 12].

5. Complexity comparison

Consideration of *q*MTD shows that linear complexity of decoding is kept. At software *q*MTD implementation the subroutine of its threshold element working which arranges practically whole decoder, occupies less than ten short lines in C++ language and provides processing simultaneously such bytes count of the received message which is supposed by architecture of used processor. Demoprogram for *q*MTD is available on web-site www.mtdbest.iki.rssi.ru. At work at a usual personal computer this demoprogram shows at code rate $R = 19/20$ and symbol error rates in *q*SC less than 10^{-2} practically optimum decoding of very long code with data rates more than 10 Mb/s, and for rather fast usual personal computers – up to 30 Mb/s. Thus the demoprogram performs all steps of data transmission: generation of an information stream, its coding, entering noise distortions and then working of discussing decoding algorithm. So real data rate of software *q*MTD decoder is even greater approximately in 1.5...2 times or more.

The complexity of *q*MTD decoders for long nonbinary codes may be compared with the complexity of decoders for RS codes which is $O(n^2)$. Various methods of increasing efficiency of decoders for RS codes, including all variations of Sudan algorithm, lead to the complexity $O(n^3)$. For codes of length $n = 30000$ symbols it leads to difference in complexity nearby $n^2 = 30000^2 \approx 10^9$, i.e. billion times. However an improvement of performance due to more complex decoding of RS codes is rather insignificant. In the best case for these algorithms the error vector weight at which reception of the correct decision is possible, increases at use of the most complex algorithm of this class less than at 4% for $R = 7/8$ and less than at 1% for $R = 19/20$. Therefore growth of correcting ability of Sudan algorithm at 4% and 1% will give nothing even for long RS codes comparable on length with the codes, chosen for *q*MTD.

At comparison the complexity of *q*MTD with the complexity of decoders for *q*LDPC codes [5–7], it appears, that the complexity of the last ones at use of enough effective decoders is $O(q \cdot \log_2 q)$ [6]. As a result the difference in complexity of *q*MTD and decoders for qLDPC codes at use, for example, four-byte symbols ($q = 2^{32}$) exceeds billion times. In [7] list message passing (LMP) decoder for *q*LDPC codes was submitted which have complexity $O(s^2)$, where $s \leq q$ is maximum list size. Using small $s << q$

essentially simplifies decoding but in this case performance is decreased significantly also. For example, LMP decoder with $s = q$ for regular qLDPC code with length 100000 symbols of size $q=2^{32}$ and code rate $R=1/2$ can work at channel SER $P_0=0.429$ and LMP decoder with $s=32$ can work only at channel SER $P_0=0.232$ [7]. It should be noted that the decoder with $s=32$ have even less error correction ability than qMTD for codes with symbols of the same size (curve *6* in fig. 1) and it is about in 1000 times more complex than the qMTD.

Conclusion

Efficiency of qMTD algorithms in symbol error rate and in complexity are at many decimal exponents better than the efficiency of Reed-Solomon codes. It is defined by effective carry of multithreshold decoding ideas on very simply organized nonbinary codes of any big length. Any other codes and decoding algorithms with similar complexity and error correction ability do not exist.

Except for natural scopes of application of simple and highly effective q-ary multithreshold decoder in communication networks, it is necessary to note good opportunities of qMTD application for coding information on disks and other storages of big volumes of information, in superbig bases of audio- and video- data with much higher levels of reliability and integrity, than it was accessible until recently, and also at updating, restoration and using of stored data. It defines all benefits of multithreshold algorithms in their applications on maintenance essentially new, on many decimal orders higher level of integrity and reliability of information storages in superbig data files of practically any structure.

Thus, this level of error correcting ability, inaccessible earlier, achievable by means of different types of qMTD algorithms allows to solve problems of high reliability maintenance for transmission and data storage without any additional completion of these algorithms or only at their insignificant adaptation to the possible additional requirements arising in large-scale digital systems.

Acknowledgements

Researches of MTD algorithms were supported by Scientific Council on the Complex Problem «Cybernetics» AS of the USSR, SRI RAS, Radio Research and Development Institute and Russian Funds of Basic Research (grants №05-07-90024 and №08-07-00078).

Additional information about different classes of MTD is located on specialized thematic bilingual web-site SRI RAS www.mtdbest.iki.rssi.ru.

The literature

1. Zolotarev V.V., Ovechkin G.V. The algorithm of multithreshold decoding for Gaussian channels. Information processes, 2008, vol. 8, №1, pp.68-93.
2. Zubarev U.B., Zolotarev V.V., Ovechkin G.V. Review of error-correcting coding methods with use of multithreshold decoders. Digital Signal Processing, Moscow, 2008, No. 1, pp.2-11.
3. Zolotarev V.V. Theory and algorithms of multithreshold decoding. M.: Radio and communication, Hot Line - Telecom, 2006, 270 p.
4. Bennatan A., Burshtein D. Design and analysis of nonbinary LDPC codes for arbitrary discrete-memoryless channels. IEEE Trans. Inform. Theory, 2006, vol.52, pp.549-583.
5. Declercq D., Fossorier M. Extended min-sum algorithm for decoding LDPC codes over GF(q) // IEEE International Symp. on Inf. Theory, 2005, pp.464-468.
6. Zhang F., Pfister H. List-Message Passing Achieves Capacity on the q-ary Symmetric Channel for Large q // In Proc. IEEE Global Telecom. Conf., Washington, DC, Nov. 2007. pp.283–287.
7. Zolotarev V.V. Multithreshold decoding over non-binary channels. Questions of radio electronics. Moscow, 1984. №12. pp.145–149.
8. Zolotarev V.V. Algorithm of symbol data correction in computing networks. Question of cybernetics, Moscow, 1985. BK-106, pp.54–62.
9. Zolotarev V.V., Averin S.V. Non-Binary Multithreshold Decoders with Almost Optimal Performance. 9-th ISCTA' 07, July, UK, Ambleside, 2007.
10. Zolotarev V.V. Generalization of algorithm MTD at non-binary codes. Mobile systems, 2007, №3, pp.39-42.
11. Zolotarev V.V. Concatenated MTD decoding schemes for large data bases. Mobile systems, Moscow, 2008, №8, pp.66-71.
12. Multithreshold decoders. Web-site of Space Research Institute R www.mtdbest.iki.rssi.ru.
13. Samoilenko S.I., Davidov A.A., Zolotarev V.V., Tretiakova E.I. Computer network. M.: Science, 1981.
14. Townsend R.L., Weldon E.J. Self-Orthogonal Quasi-Cyclic Codes. IEEE Trans., 1967, IT-13, pp.183-195.
15. Massey J.L. Threshold decoding. M.I.T. Press. Cambridge. Massachusetts. 1963.
16. Sudan M. Decoding of Reed-Solomon codes beyond the error-correction bound. Journal of Complexity, 1997, vol.13, pp.180-193.