The Performance of Multithreshold Decoders in Concatenated Schemes Over Erasure Channels

Zolotarev V.V.
Department 71
Space Research Institute of Russian Academy of Sciences
Moscow, Russian Federation
zolotasd@yandex.ru

Grinchenko N.N., Ovechkin G.V., Ovechkin P.V.
Faculty of Computer Engineering
Ryazan State Radio Engineering University
Ryazan, Russian Federation
ovechkin.g.v@rsreu.ru

Abstract—**Multithreshold decoders (MTD) of self-orthogonal codes (SOC) for erasure channels implementing error correction methods based on searching global extremum of functions in discrete spaces are considered. To increase the efficiency of erasure recoveries the concatenated codes comprising inner SOC and outer codes being simple for decoding such as parity-check codes, Hamming codes or BCH codes are offered. The usage of the codes offered provides efficient erasure recovery when operating near channel capacity with linear decoder complexity. The questions of high-throughput software MTD implementation recovering erasures using GPU are considered. The MTD versions developed are shown to be able to perform data flow decoding with the rate of several hundred MB/s using GPU.**

Keywords—error-correction coding, multithreshold decoder, self-orthogonal code, concatenated code, parity-check code, hamming code, BCH code, erasure recovering, channel capacity, GPU implementation.

I. Introduction

Error-correcting coding is widely used in communication systems to correct the errors arisen during data transmission [1]. Over the past decades efficient methods of coding and simple decoding providing the operation near the capacity of typical communication channels including the model of channel with erasures have been developed [1]. Despite its simplicity this model can be used in modeling of computer systems, data storage systems and many other systems. Besides, the decoders for this model of erasure channel turn out to be computationally simpler than the decoders for the channels with errors. Therefore, in high-rate data transmission and data storage systems a receiver doesn't use a complex error-correcting algorithm, but with the help of check sums erases certain unreliable symbols or even the whole blocks of symbols for their further recovery using error-correction code decoder. Here we should mention that nowadays for such channel different codes and methods of their decoding achieving its maximum throughput are known [2, 3, 4], but in case of finite code length their efficiency is not always the best, the decoder complexity remaining residually high.

Currently one of the best decoders from the view point of the ratio between their efficiency and implementation complexity of erasures recovery methods is multithreshold decoder (MTD) [5..9] for self-orthogonal codes (SOC) being the development of Massey threshold decoder [10]. A given method with linear complexity provides close to optimum decoding of correctly chosen SOC in sufficiently big range of code and channel parameters [11].

The rest part of the paper is arranged as follows. Section 2 considers several questions of multithreshold decoding over erasure channels and offers concatenated codes allowing to increase the efficiency of erasure recoveries, a lower bound for erasure non-recovery probability is given. Section 3 performs the analysis of offered and constructed codes efficiency. Section 4 considers the aspects of high-throughput software MTD implementation. Section V concludes this paper.

II. An Improving Performance of Multithreshold Decoder over Erasure Channels

The operation of SOC encoder and their multithreshold decoder for binary and q-ary channels with errors are explained in a rather detailed way in works [5, 7, 8, 12]. In erasure channel the algorithm of encoder operation is practically the same. The operation of MTD being able to recover erasures in such channel is as follows [5, 11]. While calculating syndrome symbols the erased information and parity-check symbols have no influence on syndrome symbol value. For each syndrome symbol the number of erased information and parity-check symbols included in its formation is memorized. In the process of erased information symbol decoding among all syndrome symbols relating to it we search the one that required only erased symbol for its formation. The value of this given syndrome symbol leads to the recovery of decoded erased information symbol requiring also to change the values of all syndrome symbols relating to it and to reduce by one the number of erased symbols necessary for the formation. Next we move to the other erased information symbol. Erased information symbols recovery will continue till the moment when decoding iteration requires no erased symbol recovery.

It should be noted that the principles of MTD operation while using binary and q-ary symbols in erasure channels turn out to be similar, but instead of addition by module 2 q-ary symbols use addition by module q. Here the dependences of symbol (binary or q-ary) non-recovery probability from symbol erasure probability in a channel for binary and q-ary MTD using similar generator polynomials coincide. Therefore, for MTD while discussing the characteristics we shall use the symbols understood both as binary and q-ary ones.

The efficiency of MTD application in erasure channels was discussed in detail in [5, 11]. Given works show the peculiarity of MTD and SOC used together with them being the fact that the codes with higher code distance are capable to provide lower erasure probability in comparison with the codes having lower code distance [11]. This fact is related to low erasure probability in a channel. In case of high channel erasure probability the codes with high code distance seem to be unable to recover them. This fact significantly complicates the possibility to receive low decoder erasure probability in case of high noise in the channel. To solve this problem we can use concatenated schemes [13] which use SOC with low code distance in inner concatenation, and outer concatenation needs simple for decoding high-rate code, e.g. parity-check code, hamming code or BCH code. MTD for inner code is supposed to operate practically the same way as optimum decoder and to sufficiently reduce erasure probability in comparison with channel erasure probability. Further, outer code decoder will allow reducing erasure probability to a greater degree providing the required decoder erasure probability even in case of higher probability of erasure in a channel.

Next we consider the principles of concatenated scheme organization where SOC decoded with MTD is used as inner, and parity-check code, Hamming code or BCH code are used as outer (Fig. 1). While decoding such code first SOC decoding with MTD is performed, next outer code decoding is made. This process can be repeated many times allowing us to improve the decisions of decoder after each iteration.

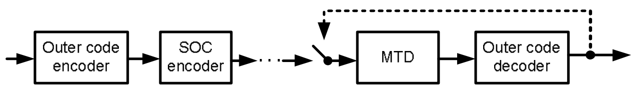

Fig. 1. Scheme of concatenated code encoding and decoding

Further we shall perform analytical evaluation of the offered scheme efficiency assuming that given codes form a product code. According to our speculations SOC with code distance d_{in} is used as inner, and the code with length n_{out} and code distance d_{out} is used as outer. If we suppose that in case of chosen erasure probability P_e in a channel MTD performs close to optimum decoding, then the MTD erasure probability can be evaluated as

$$P_{MTD} = P_e^{\,d_{in}} . \quad (1)$$

Outer code with code distance d_{out} allows to recover $d_{\text{out}}-1$ erasures in a block of length n_{out}. If the block has more erasures they are not recovered. Consequently, the outer decoder erasure probability is evaluated as

$$P_{out} = \sum_{k=t+1}^{n_{out}} \frac{k}{n_{out}} C_{n_{out}}^{k} P_{MTD}^{k} (1-P_{MTD})^{n_{out}-k} . \quad (2)$$

III. Performance Analysis

Fig. 2 represents the dependences of decoder erasure rate from the channel erasure probability for different outer codes. Here SOC with code rate R_{in}=1/2, code distance d_{in}=9 and length n_{in}=10200 symbols was used as inner code. The graph of erasure probability dependency after inner code decoding from channel erasure probability is shown by curve *1*. Used as outer parity check code of length 255 symbols and recovering one erasure we receive the characteristics represented by curve *2*, used as outer parity check code of length 32 symbols and recovering one erasure we receive the characteristics represented by curve *3*, used as outer Hamming code (255, 247) recovering two erasures we receive the characteristics represented by curve *4*, used as outer BCH code (255, 239) recovering four erasures we receive the characteristics represented by curve *5*. The dotted lines in the figure show the lower bounds of decoder erasure probability. Here we should note that by applying outer codes we manage to receive 1…5 decimal orders less decoder erasure probability in the area of MTD efficient operation. It’s also worth mentioning that lower erasure probability evaluations received after decoder for the concatenated schemes offered turn out to be sufficiently accurate.

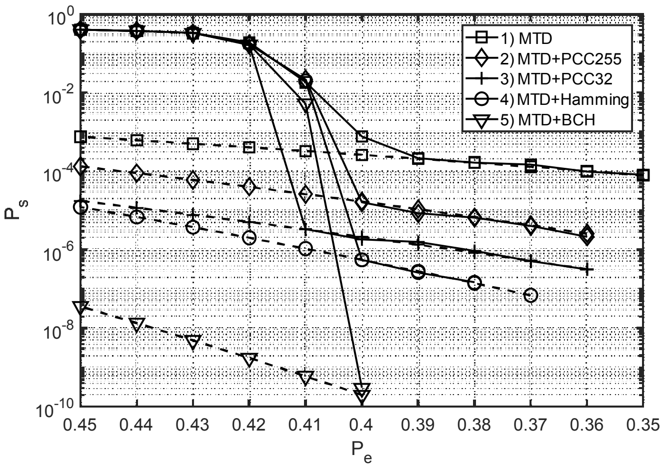

Fig. 2. Characteristics of MTD based erasure recovery methods

Below we shall consider the efficiency of best known erasure recovery methods illustrated by the first two graphs in fig. 5. They shows the characteristics of the best low-density parity-check codes (LDPC) of length 524288 (curve *1*) [3], accumulate-repeat-accumulate codes (ARA) of length 65536 concatenated with outer parity-check code (curve *2*) [4], as well as known MTD for block and convolutional codes with code rate R=1/2 (curves *3* and *4*) [11]. Further we shall consider our results. One of the best block codes constructed is SOC with R=1/2 and d=19. Its characteristics for 65 decoding iterations are shown by curve *5*. It should be mentioned that sufficient increase in efficiency of known block MTD is achieved. An outer parity-check code of length 50 symbols with iterative decoder being used with the SOC given provides decoder erasure rate a little more than 10^{-11} (curve 6) in case of channel erasure probability 0,47. One of the best constructed convolutional codes is SOC with R=1/2 and d=21. Constraint length of the code was about 200000 symbols, 100 iterations were made during its decoding. The efficiency of such MTD is represented by curve *7*. This MTD successfully recovers the flow of erased symbols with channel erasure probability P_e=0,48 till the level of decoder erasure rate $P_s=3\cdot10^{-7}$. Using outer parity check code and iterative decoder with this code one can provide decoder erasure rate less than 10^{-11} (curve 8) for channel erasure probability 0,48. In this case the characteristics will be even better than the characteristics of known methods.

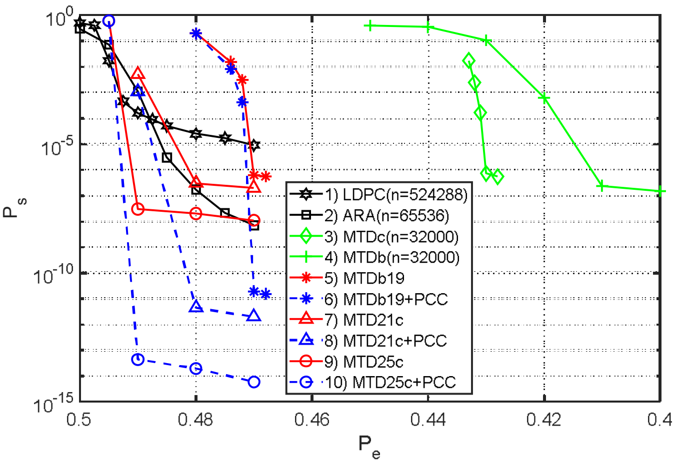

Fig. 3. Characteristics of modern methods to recover erasures for the codes with code rate 1/2

One more result received by our team of scientists is represented by curve *9*. It corresponds to MTD for a new code with R=1/2 and d=25. As the graph shows, this MTD operates efficiently for channel erasure probability P_e=0,49 and achieves decoder erasure rate $3\cdot10^{-8}$ which is an impossible task for other erasure recovery methods. MTD scheme under consideration is related to standard, earlier described versions of non-concatenated decoders actively using the principle of divergence [14, 15] being applied several times while SOC construction. The number of iterations used was not more than I=200 for P_e=0,49, and complete delay of decoder decision was less than 2 million symbols. Using outer parity check code and iterative decoder with this code given in case of channel erasure probability 0,49 the decoder erasure probability less than 10^{-13} (curve *10*) can be provided.

It should also be noted that by applying concatenated codes with more powerful outer codes (Hamming, BCH codes) the decoder erasure probability in the area of efficient MTD operation can be reduced, but this leads to a slight increase of code redundancy and decoding complexity.

IV. Software MTD Implementation

For high-throughput software MTD implementation much attention must be paid to computation parallelization capabilities. Currently practically all consumer computers as well as modern mobile phones feature multicore CPUs. But in case of software MTD implementation for erasure recovery the fastest decoder software versions provide the throughput only about 15 MB/sec using 6 cores of Core i7 processor for simultaneous parallel processing of six blocks received. A prospective direction to accelerate the operation of MTD software versions is their implementation on graphical processing units (GPU) using CUDA architecture [16]. A distinctive feature of GPU application in comparison with CPU application is the fact that modern GPU have thousand cores allowing to perform in parallel several thousand of operations of the same type. One of the main ways to create a high-throughput decoder is the placement of all arrays used by encoder and decoder in shared and constant memory, ensuring conflict-free access to the elements of these arrays as well as simultaneous decoding of several symbols inside one message received from the channel by GPU threads used.

Next the aspects of software MTD version implementation using graphical card NVidia GeForce GTX 1060 are considered. The results of the work performed for SOC with R=1/2, n=1600 one byte symbols and d=5 (with checks located in positions 0, 109, 295 and 372) on the basis of CUDA architecture allowed us to develop block MTD recovering erasures which uses only register variables and only 4,7 Kb of shared memory intended for storing data of information and syndrome registers. This gives the possibility in each of 10 multiprocessors (SMM) to perform simultaneous decoding of 20 received blocks, i.e. total simultaneous decoding of 200 received messages is performed. Fig. 4 represents the scheme of multiprocessor usage offered by the authors.

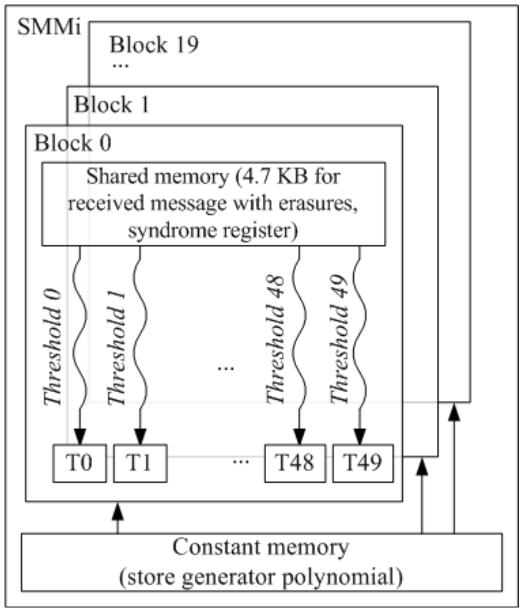

Fig. 4. Scheme of multiprocessor usage

To increase decoding throughput polynomials for SOC used were selected in such a way that the positions of checks are not less than 50 positions apart. This allows us to use 50 threads for one block decoding in order to correct simultaneously 50 symbols with conflict-free access and record to shared memory. The scheme of simultaneous operation of 50 MTD threshold elements for the code is given in Fig. 5. In the first cycle threads 0, 1, …, 49 are responsible for decoding of 750, 751, …, 799 symbols of information register. At the same time, due to thread synchronization only 50 consecutive symbols of shared memory are accessed each time eliminating the conflicts of memory banks. In the next cycle the 50 threads used perform parallel decoding of 700, 701, …, 749 symbols of information register, etc. Therefore, thread 0 performs the decoding of 799, 749, 699, …, 99, 49 symbols of the received message, thread 1 – 798, 748, 698, …, 98, 48, and thread 49 – 750, 700, 650, …, 50 and 0 symbols of MTD block.

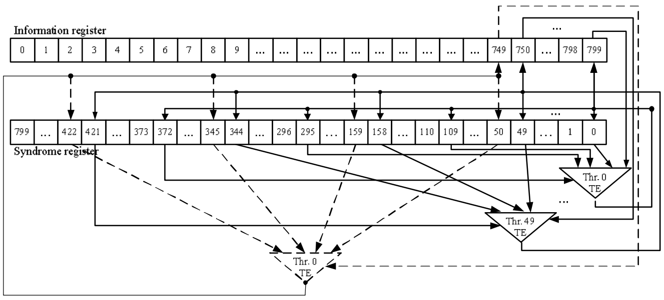

Fig. 5. Scheme of MTD using 50 parallel threshold elements

According to the table 1 which presents the results of decoder operation throughput evaluation, the application of the methods offered provided total decoding throughput for the whole system being equal to 270 MB/s (more than 2 Gbit/s) for the code with R=1/2, block length 1600 symbols, d=5 and 20 decoding iterations. For the code with code length 4000 symbols, d=9 during 20 iterations MTD provides throughput being equal to 150 MB/s. Therefore, software MTD versions can be used in high throughput software oriented systems based on graphical processors.

Table 1. Results of decoder throughput evaluation

Code and decoder parameters	Number of blocks decoded simultaneously	Through-put
d=5, n=1600, it=20	200	270 MB/s
d=7, n=2200, it=20	140	225 MB/s
d=9, n=4000, it=20	80	150 MB/s

V. CONCLUSION

The paper presents the main results of the development of simple for implementation and efficient multithreshold decoders for erasure channels which can be used in telecommunication systems as well as data storage systems. To increase MTD efficiency the authors offered concatenated coding schemes based on simple for decoding outer codes and inner SOC the application of which allows reducing erasure probability for a few decimal orders after decoding without sufficient complexity increasing in comparison with MTD.

The work also considers the features of software MTD implementation using GPU. The decoders developed are shown to be capable to ensure input thread processing with the throughput from 150 to 270 MB/s with the help of NVidia GeForce GTX 1060. Such rate allows applying MTD in high throughput software oriented systems.

Sufficient amount of information about MTD is presented in our web-sites [15].

ACKNOWLEDGMENT

The research is carried out due to the support of the Ryazan State Radio Engineering University and Russian Foundation of Basic Research (grant №18-07-00525).

REFERENCES

[1] Channel Coding: Theory, Algorithms, and Applications / M. Fossorier, D. Declerq, E. Biglieri and others. – Academic Press Library in Mobile and Wireless Communications, Elsevier, 2014. – 690 p.

[2] Arıkan E. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels, IEEE Trans. Inf. Theory, vol. 55, no. 7, pp. 3051-3073, Jul. 2009.

[3] Pfister H.D., Sason I., Urbanke R., Capacity-achieving ensembles for the binary erasure channel with bounded complexity, IEEE Trans. Inform. Theory, 2005, vol. 51, no. 7, pp. 2352–2379.

[4] Pfister H.D., Sason I., Accumulate-Repeat-Accumulate Codes: Systematic Codes Achieving the Binary Erasure Channel Capacity with Bounded Complexity, IEEE Transactions on Information Theory, 2005.

[5] Zolotarev V.V., Zubarev Y.B., Ovechkin G.V. Optimization Coding Theory and Multithreshold Algorithms // Geneva, ITU, 2015, 159p.

[6] Zolotarev V.V., Ovechkin G.V., Chulkov I.V., Ovechkin P.V., Averin S.V., Satibaldina D.Zh., Kao V.T. Review of achievements in the optimization coding theory for satellite channels and Earth remote sensing systems: 25 years of evolution, Sovremennye Problemy Distantsionnogo Zondirovaniya Zemli iz Kosmosa, 2017, vol.14, no.1, pp.9-24.

[7] Zolotarev V., Ovechkin G., Issainova A., Satybaldina D., Tashatov N. Effective multithreshold decoding algorithms for wireless communication channels (2016) Application of Information and Communication Technologies, AICT 2016 - Conference Proceedings.

[8] Zolotarev V.V., Ovechkin G.V. Algorithm of multithreshold decoding for Gaussian channels // Automation and Remote Control. 2008. no.69, pp.1086–1100.

[9] Ullah M.A., Omura R., Sato T., Ogivara H. Multi-Stage Threshold Decoding for High Rate Convolutional Codes for Optical Communications. AICT 2011: The Seventh Advanced international Conference on Telecommunications, pp. 87-93.

[10] Massey J. Threshold decoding. M.I.T. Press, Cambridge, 1963.

[11] Grinchenko N., Gromov A., Ovechkin G. Improving performance of multithreshold decoder over binary erasure channel // 6th Mediterranean Conference on Embedded Computing, MECO 2017 - 2017.

[12] Zolotarev V.V., Ovechkin G.V. Efficient Multithreshold Decoding of Nonbinary Codes. Journal of Communications Technology and Electronics, 2010. Vol. 55, No. 3, P. 302–306.

[13] Zolotarev V.V., Ovechkin G.V., Ovechkin P.V., Pilkin A.N. The Application of Modulo q Check Codes to Increase the Efficiency of Non-Binary Multithreshold Decoders Over q-ary Symmetric Channel // 2018 Moscow Workshop on Electronic and Networking Technologies (MWENT). Proceedings. Moscow, March 14-16, 2018.

[14] Zolotarev V., Grinchenko N., Lotsmanov A., Ovechkin G. Developing the Principle of Divergent Coding for Gaussian Channels // 7-th Mediterranean Conference on Embedded Computing MECO'2018, Budva, Montenegro.

[15] Web sites www.mtdbest.iki.rssi.ru and www.mtdbest.ru.

[16] Zolotarev V., Ovechkin G., Ovechkin P., Satybaldina D., Tashatov N., Sankibayev D. High Throughput Software Multithreshold Decoder on GPU // 3rd int.conf. on mathematics and computers in sciences and industry - MCSI 2016. Chania, Greece, August 27-29, 2016.