# The algorithm for determining the errors multiplicity by multithreshold decoding of iterative codes

**P.P. Urbanovich[1,2], V.M. Vitkova[2], D.M. Romanenko[2]**
[1]*Lublin Catholic University, Lublin, Poland, upp@rambler.ru*
[2]*Belarusian State Technological University, Minsk, Belarus, mvitkova@bk.ru*

Modeling of multithreshold decoding of three-dimensional iterative codes [1] with five parities allowed to identify some features of multiple errors correction. These features allow to determine the multiplicity of corrected errors by specific algorithm directly in the decoding process, i. e. by transmission of real traffic over the data channel.

It is necessary to introduce the following notation to describe this algorithm: $i$ - stage number at which a check is carried out; $j$ - bit number, which is checking; $t_{i,j}$ - number of parities, indicating about error in the checking bit; $T_i$ - threshold value at the certain decoding stage; $c$ - number of inverted bits; $n = k + r$, where $n$ - length of the code word, $k$ - number of information bits, $r$ - number of parity bits; $f$ - parities availability flag, which indicating error in checking bit; $m$ - error multiplicity; $s$ - number of decoding stage.

The developed algorithm can be divided into the following steps (for example, decoding the code with 5 linearly independent parities).

Step 1: a) assign $i = 1$ – start checking information bits in the received code sequence at the first decoding stage; assign $j = 1$, because the first bit of the code message is checking;

b) assign $T_i = 5$ – threshold at the first decoding stage, because the iterative code with five parities is using;

c) assign $c = 0$; $k = 64$ – the length of information message; $f = 0$ – at the first step no parities are indicating about error.

Step 2: count the parities $t_{i,j}$ of $j^{th}$ transmitted symbol at the $i^{th}$ decoding stage, which indicated about error in the checking bit.

Step 3: a) if $t_{i,j} > 0$, then variable $f$ assign 1 ($f = 1$), else go to step 4;

b) if $t_{i,j} \geq T_i$ (threshold is reached), then invert bit and increase the variable $c$ by 1 ($c = c + 1$), else go to step 4.

Step 4: turn to the analysis of the next received code symbol, i. e. if $j < n$, then $j = j + 1$ – go to subparagraph $c$ of the 1st step.

Step 5: a) if $i < s$, then $i = i + 1$ – go to the next stage of decoding, else – the decoding process is completed and go to step 6;

b) assign $j = 1$, because checking at the next stage of decoding begins from the first bit; asssign $f = 0$; $t_{i,j} = 0$ and return to step 2.

Step 6: if $f = 0$, then assign $m = c$, otherwise the conclusion of the impossibility of determining the multiplicity of errors during the decoding process is done.

The algorithm allows to determine the errors multiplicity if no parity does not show the error of even one bit at the last (third) decoding stage. In this case multiple error $m$ equals $c$ – the count of inverted bit. Besides, if there was inverted consecutive bits, it means, that the modular error, which multiplicity is equals the variable $m$, was corrected. Else the error was independent. It should be noted that some bits of multiple modular errors can be detected and corrected at the later stages of decoding.

## References

[1] Urbanovich P.P., Patsei N.V., Romanenko D.M., Shiman D.V. Multilevel turbocoding schemes on the basis of twodimensional linear iterative codes withdiagonal checks, Przegląd elektrotechniczny, 84 (2008), No.3, p. 152-154.