Calculation of reliability of information symbols in perforated codes

Alexander A Sidorenko

International Academy of Informatization, 3 Leningradsky ave., Moscow, Russia

E-mail: likotasa@mail.ru

Abstract. Turbo-codes came closest to Shannon’s limit. One of the important steps in the operation of a turbo-code decoder is to calculate the reliability of information symbols. The article describes in detail the process of calculating the reliability of information symbols in accordance with the SOVA algorithm for convolutional code with perforation of code bits and without perforation. For a specific example of a received code sequence containing an error, numerical calculations of the reliability of information symbols for a convolutional code with perforation of code bits and without perforation are performed. The results are compared.

Errors can occur when storing data and transmitting information over communication networks. Error correction is an important task at many levels of working with information, for the solution of which, error-correcting codes with error correction are used among other methods. Noise-resistant codes are quite diverse. In convolutional coding, \( n \) binary symbols obtained as a result of encoding depend not only on the next group of \( k \) binary symbols of the information message, but also on \( K-1 \) of the previous groups.

For decoding convolutional codes, the Viterbi algorithm, which performs decoding to the maximum likelihood, is widely used. Based on the results of the implementation of the algorithm, the code sequence closest to the accepted one is determined. For the Viterbi algorithm, there are two main options for determining such a sequence: by minimizing the Hamming or Euclidean distance. In the decoding process, in accordance with the classical Viterbi algorithm [1], iterative movement occurs along the lattice, the nodes of which are the values of the memory cells of the encoder from left to right. For each possible branch of the transition between nodes, a metric is calculated - a number characterizing the degree of difference between the characters received at the decoder input (for a given iteration) and the characters generated by this branch (branch value). Paths are formed from the branches corresponding to possible variants of the code sequence. For each possible path, the path metric is calculated as the sum of the metrics of the branches included in the path. At each iteration, starting from the third, two paths enter the node, and one with the lower metric comes out. Based on the results of the passage through the bars from left to right (direct passage), the final surviving path (having the smallest metric of the path) is selected, which corresponds to the most probable code sequence that uniquely determines the values of information symbols.

In the last decade, turbo codes have been actively introduced into various digital data transmission systems [2]. These codes came closest to the Shannon limit [3, 4]. One of the important steps in the operation of a turbo code decoder is to calculate the reliability of information symbols. There is a
modified Viterbi algorithm with soft-output viterbi algorithm (SOVA) [1]. The reliability of \( u_i \) information symbol is calculated as the logarithm of the likelihood relationship:

\[
\Lambda(u_i) = \log \frac{Pr(u_i=1|r)}{Pr(u_i=0|r)}
\]  

where \( r \) is an accepted sequence. By simple transformations \([m - c] \), formula (1) can be reduced to the following form:

\[
\Lambda(u_i) = M_i(0) - M_i(1).
\]  

That is, for \( i \) section of the lattice, the reliability of \( u_i \) information symbol is equal to the difference between \( M_i(0) \) minimum path metric for \( u_i=0 \) case and \( M_i(1) \) minimum path metric for \( u_i=1 \) case. The essence of SOVA algorithm [5] is the calculation of the reliability of the information symbol by the formula (2). The “+” or “-” sign of the reliability of the information symbol calculated by the formula (2) corresponds to the adoption of a tough decision on the value of the information symbol, and the absolute value characterizes the degree of reliability of the tough decision. In this case, the “+” sign corresponds to the adoption of a tough decision on decoding “1”, and “-” on decoding “0”.

The SOVA algorithm can be used in binary or multi-level modulation, for decoding data blocks or in a continuous sequence (in the sliding window mode), independently or as a component of a turbo code decoder [6]. At relatively low complexity, the efficiency of the SOVA algorithm asymptotically approaches the efficiency of optimal decoding algorithms with the maximum a posteriori probability while reducing the probability of an erroneous bit in the data transmission channel [7, 8].

Let us consider the calculation of the reliability of information symbols for the case of the simplest unsystematic convolutional code, the encoder of which is shown in figure 1. For other types of convolutional codes, the process of calculating the reliability of information symbols will be similar to that discussed below.

![Figure 1. Non-systematic convolutional encoder.](image1)

Figure 2 shows a fragment of the lattice corresponding to one iteration of decoding on the left. From each node of the lattice, the lines show two allowed branches corresponding to the information symbols “1” or “0”. The upper branches emerging from the node correspond to the information symbol “0”, and are shown in the center of figure 2. The lower branches correspond to the information symbol “1”, and are shown on the right of figure 2.

![Figure 2. Allowed transitions between lattice nodes (branches).](image2)
All branches are part of the various permitted paths. To calculate the reliability of the information symbol corresponding to this iteration, it is necessary to find the minimum metric for paths containing branches corresponding to the information symbol "0" (that is, $M_i(0)$), and find the minimum metric for paths containing branches corresponding to the information symbol "1" (that is $M_i(1)$), and substitute the found values into formula (2).

Let the encoder be encoded by 110100 information sequence, as a result of which 11 01 00 10 11 code sequence is obtained. Let us consider the calculation of the reliability of information symbols for the case of a hard decision by the detector (the operation of the algorithm for the case of a soft decision by the detector will differ only in the metric calculation rule). Let the sequence 11 01 00 00 10 11 be detected, that is, there is one error bit (underlined). A direct decoding pass with the calculation of Hamming metrics of allowed paths is shown schematically in Figure 3. The following notation is used on the decoding scheme: pairs of numbers on the branches correspond to a pair of bits generated by the encoder (with such a transition between nodes), numbers to the left of the lattice node are the values of the metrics of the paths entering the node, the number to the right of the lattice node is the metric value of the path exiting the node. The decoded sequence is shown at the top of the circuit and italicized. The final surviving path is highlighted by a solid line.

Figure 3. A direct encoding passage.

When decoding a final sequence encoded by a convolutional code, the last bits of the sequence are the least error protected. To increase the security of the last bits, termination bits [1] are added to the information sequence, which reset the encoder memory cells to zero (for the considered information sequence, these are the last two bits). When the final state of the memory cells is known, this simplifies the decoding process, making it possible to exclude from consideration a number of impossible paths at the end of the lattice. Figure 4 repeats figure 3, with impossible paths eliminated at the end of the lattice, and on each branch the value of its metric is indicated (instead of the values of the generated bits). The return passage is carried out similarly to the direct passage, and is schematically shown in figure 5. The return passage begins on the right side of the lattice, which means the numbers to the right of the lattice node are the metrics of the paths entering the node, the number to the left of the lattice node is the metric value of the path leaving the node.

Figure 4. A direct encoding passage. Branch metrics.
As it was mentioned earlier, in the course of a direct passage, the path having the smallest metric is determined. We call this path the final survivor. If for this iteration the final surviving path corresponds to the information bit “0”, then the value of $M_i(0)$ is equal to the metric of the final surviving path, and only $M_i(1)$ has to be found. Similarly, if for a given iteration the final surviving path corresponds to the information bit “1”, then the value of $M_i(1)$ is equal to the metric of the final surviving path, and only $M_i(0)$ has to be found.

Let us consider in detail the calculation of the reliability of the third information symbol. At the third iteration of decoding, the final surviving path passes from node “11” to node “01” and corresponds to the value of the information symbol “0” (see figure 7 on the left). The metric of the final surviving path is 1, then $M_3(0)=1$. We will then consider finding the minimum path metric corresponding to the information symbol “1”. Such paths contain the branches shown in the center of figure 7.

The metric of the path containing one or another branch is equal to the sum of three terms: the metric of the direct path from the beginning of the lattice to this branch ($MF$), the metric of the branch itself and the metric of the return path from the end of the lattice to this branch ($MB$). Figure 7 (on the right)
shows the calculation of the minimum metric of a path that includes a branch from node “01” to node “10”. Similarly, to the considered one, one can calculate the metrics of three other paths corresponding to the information symbol “1”. The data for the calculations and the calculation results of the path metrics are shown in table 1. Having determined the minimum value of the path metric from the table, we obtain \( M_3(1) = 4 \).

### Table 1. The data for the calculations and the calculation results of the path metrics.

| Branches, included into the path | MF | Branch metrics | MB | Branch metrics |
|---------------------------------|----|---------------|----|---------------|
| «00» - «10»                     | 3  | 2             | 3  | 8             |
| «10» - «11»                     | 3  | 1             | 3  | 7             |
| «01» - «10»                     | 2  | 0             | 3  | 5             |
| «11» - «11»                     | 0  | 1             | 3  | 4             |

Having substituted \( M_3(0) \) and \( M_3(1) \) into (2) we will calculate the reliability of the third decoded information symbol:

\[
Λ(u_3) = M_3(0) - M_3(1) = 1 - 4 = -3. \tag{3}
\]

The reliability values of the remaining symbols, calculated similarly, are shown in table 2. The calculation of the reliability of the last two symbols (terminating symbols) is simplified since there are no branches corresponding to the information symbol “1”. Terminating characters in this case have absolute reliability. It should also be noted a remarkable fact that working in the considered example with hard detector solutions, soft decisions were obtained regarding the values of information bits.

An urgent task is to adapt the corrective ability of the error-correcting code (the ability of the code to correct errors) to changes in the characteristics of the data transmission channel. This can be realized, for example, by excluding (puncturing) certain code bits from the stream of transmitted data (perforation of the code), since the potential corrective ability of the code depends on redundancy.

### Table 2. The reliability values of the remaining symbols.

| \( i \) | 1  | 2  | 3  | 4  | 5  | 6  |
|--------|----|----|----|----|----|----|
| \( Λ(u_4) \) | 3  | 4  | -3 | 4  | -∞ | -∞ |

Let us consider the calculation of the reliability of information symbols for the case of a perforated code sequence. There are various rules for puncturing bits when using convolutional codes [9]. Let every fourth bit in the previously considered code sequence be punctured, and the detected sequence also correspond to the previously considered. Then, taking into account the perforation, the sequence 11 0X 00 0X 10 1X is received at the decoder input, where the X symbol denotes punctured bits that are not involved in the calculation of the metric. Figure 8 schematically shows a direct decoding pass of a perforated sequence. Symbols are similar to those used for the scheme in figure 4. The final surviving path corresponds to that found earlier for the received sequence without perforation. Figure 9 shows the results of both passes (symbols are similar to those used in the scheme in figure 6). Using the metrics obtained as a result of the direct and reverse pass, it is possible to calculate the reliability of information symbols. The calculation results are shown in table 3.
Figure 8. A direct encoding passage. Perforated code sequence.

The strict values of the information bits obtained for code sequences with and without perforation are the same, and the reliability of the values for the case of a perforated sequence is lower, which is consistent with a decrease in the corrective ability that arose due to code perforation.

Figure 9. Direct and return encoding passages. Perforated code sequence.

Table 3. The results of calculating the reliability of information symbols.

| i   | 1  | 2  | 3  | 4  | 5  | 6  |
|-----|----|----|----|----|----|----|
| $A(u_i)$ | 2  | 2  | -2 | 2  | $-\infty$ | $-\infty$ |
| Decision on the value of the information bit | 1  | 1  | 0  | 1  | 0  | 0  |

The calculation of the reliability of information symbols is a necessary step in the operation of a turbo-code decoder. Reliability of information symbols can be obtained using the modified Viterbi algorithm with soft output, while the algorithm can work with both hard and soft detector solutions. The algorithm without significant changes works with perforated code sequences, while the expected decrease in the reliability of hard decisions is observed.

References
[1] Yeung R W 2008 Information theory and network coding (Springer)
[2] Richardson T and Urbanke R 2008 Modern coding theory (Cambridge University Press)
[3] Berrou C and Glavieux A 1996 Near optimum error correcting coding and decoding: Turbo-codes IEEE Transactions on Information Theory 44(10) 1261-71
[4] Berrou C, Glavieux A and Thitimajshima P 1993 Near Shannon limit error-correcting coding and decoding: Turbo-codes (1) IEEE International Conference on Communications 2 1064-70
[5] Vucetic B and Yuan J 2000 Turbo Codes: Principles and Applications (Springer)
[6] Semenov S and Trofimov A 2011 Convolutional codes and turbo-codes. Modulation and Coding Techniques in Wireless Communications (St. Petersburg: St. Petersburg State University of
Aerospace Instrumentation) pp 161-205

[7] Viterbi A J 1998 An Intuitive Justification and a Simplified Implementation of MAP Decoder of Convolutional Codes *IEEE Journal on Selected Areas in Communications* **16**(2) 260-4

[8] Dumer I 1996 Suboptimal decoding of linear codes: partition technique *IEEE Transactions on Information Theory* **42** 1971-86

[9] Yasuda Y, Kashiki K and Hirata Y 1984 High-Rate Punctured Convolutional Codes for Soft Decision Viterbi Decoding *IEEE Transactions Communications* **32**(3) 325-8