UDC 621.391

IMPROVED SOFT DECODING METHOD APPLIED TO DATA RECEIVED FROM TELECOMMUNICATION CHANNEL

Valentin Pocotilenco, Ph.D. Ghenadie Bodean, Ph.D. Tatiana Shestakova

Technical University of Moldova
Moldova, Chisinau
poco@mail.utm.md

Errors occuring during the signal transmission through telecommunication channels create the problem that can be solved on different stages of data lifecycle, beginning with data generation and transmission through the communication channel on the transmitter and ending with data decoding and processing on the receiver. This paper is focused on a new improved sliding-sweep probabilistic algorithm of linear non-binary codes decoding, which simplifies the a-posteriori probabilities (APP) calculus. It is also proposed that decision-making should be based on log-likelihood ratio (LLR).

Keywords: signal processing and transmission, trellis, soft-decision decoding.

One of the main aims of university nano-satellite project "SATUM" is image capturing of terrestrial surface of Moldova and transmission of the captured images to the ground station in real-time mode. Each captured image is unique and one of the problems is that the retransmission of lost or defected images is not possible. Generally, for processing of unique data the soft based decoding is used. For different groups of codes specific soft decoding algorithms can be used. For example, convolutional codes are decoded with the use of Viterbi algorithm. For linear codes the algorithm based on syndrome metrics (*one-sweep*) is applicable [1].

The key moment in the *one-sweep* algorithm is to use reduced vectors, where the excluded component is assumed to be erroneous. In order to calculate probabilities of the reduced vectors using the improved *one-sweep* [2] a large amount of computing power is also required. Based on the *one-sweep* algorithm [1, 2], applied for soft-decoding linear codes [3], the matrices-based *sliding-sweep* calculus method for the codewords metric and "reduced" probabilities is proposed.

Soft decision decoding using the *sliding–sweep* method

The *one-sweep* algorithm is based on iterative calculation of APP [4], using following expression:

$$P(c_n = \alpha \mid \mathbf{r}, \mathbf{c} \in C) = \frac{P(\mathbf{r}_n^{\perp} \mid c_n = \alpha, \mathbf{c} \in C) \cdot P(r_n \mid c_n)}{\mid Б \mid \cdot \mu(0, N)}, \quad (1)$$

$$\mathbf{r}_n^{\perp}(r_1, r_2, \ldots, r_{n-1}, r_{n+1}, \ldots, r_N), \quad (2)$$

where $P(\mathbf{r}_n^{\perp} \mid c_n = \alpha)$ are conditional probabilities of receiving **r** when c is transmitted, which can be calculated using (2); $P(r_n \mid c_n = i)$ are statistic values of communication channel; $\mu(0, N)$ is value of metric; $|A|$ is input symbol degree.

The output of *one-sweep* algorithm is

$$P(c_n = \alpha \mid \mathbf{r}) = \max_{\alpha \in A} \{P(c_n = \alpha \mid \mathbf{r}, \mathbf{c} \in C)\}. \quad (3)$$

To simplify the APP computation, in (3) we can use the following expression instead of (1):

$$P(c_n = \alpha \mid \mathbf{r}, \mathbf{c} \in C) \cong P(\mathbf{r}_n^{\perp} \mid c_n = \alpha, \mathbf{c} \in C) \cdot P(r_n \mid c_n \neq \alpha). \quad (4)$$

In the proposed formula (4) only codewords C metric is used.

In order to decrease dimension of the trellis (**T**) diagram, used for probabilities $P(\mathbf{r}_n^{\perp} \mid \bullet)$ calculus, trellis codewords **T**(C)⊂**T** can be generated. To generate the codewords of the trellis **T**(C), a *back feed* (BF) algorithm is developed (fig. 1), where $\mathbf{W}_{(\bullet)}$ are weighted adjacent matrices with elements w_{ij} defined as

$$\mathbf{W} = \mathbf{W} \ \mathbf{mask} \ \mathbf{X}, \tag{5}$$

where formation rule for **W** is

$$w_{ij} = w_{ij} \bigcup_m x_j = x_i \bigcup_m w_{ij} = \begin{cases} w_{ij}, \text{ when } x = 1, \\ \varnothing, \text{ when } x = 1. \end{cases} \tag{6}$$

Commutative operation with matrix multiplying will be defined to the right-hand:

$$\mathbf{X} = < x_j >_{1\times Q} = \mathbf{X} \otimes \mathbf{W}, \text{ where } x_j = \vee_i \left(x_i \underset{\mathrm{m}}{\wedge} w_{ij} \right), \tag{7}$$

and the left-hand:

$$\mathbf{W} \otimes \mathbf{X}^T = \mathbf{X} = [x_i]_{Q\times 1}, \text{ where } x_i = \vee_i \left(w_{ij} \underset{\mathrm{m}}{\wedge} x_j \right), \tag{8}$$

where $\underset{m}{\wedge}$ is logical operation AND extension, defined as

$$w \underset{\mathrm{m}}{\wedge} x = x \underset{\mathrm{m}}{\wedge} w = \begin{cases} 1, \text{ if } x \neq 0 \ \& \ w \neq \varnothing, \\ 0, \text{ otherwise.} \end{cases} \tag{9}$$

The objective of BF-algorithm is elimination from **T** the extra-edges that are associated with erroneous transitions. After reducing the full syndrome trellis **T** to truncated **T**(C) the *sliding sweep* algorithm (fig. 2) is applied to the new set **W**=($\mathbf{W}_1,\ldots,\mathbf{W}_N$), where N is the codeword length.

Backward algorithm:
input: $\mathbf{W}_{K+1},\ldots,\mathbf{W}_N$.
1. **setup** $\mathbf{X}_n$=<$x_j(n)$>, where $x_{j=0}$=1 and $x_{j\neq 0}$=0, 1≤j≤Q.
2. **for** n **in** N **downto** K+1 **do**
   $\mathbf{W}_n := \mathbf{W}_n$ **mask** $\mathbf{X}_n$; $\mathbf{X}_{n-1} := \mathbf{W}_n \otimes \mathbf{X}_n^T$;
output: updated $\mathbf{W}_{K+1},\ldots,\mathbf{W}_N$.

Fig. 1. Backward algorithm

The *sliding-sweep* algorithm is based on matrix masking operations. Also, this algorithm contains a forward and backward iterations that diverge (like a *sliding* sash) from the position n of the error. After performing the matrix masking a reduced set $\mathbf{w}_n^{\perp} = (\mathbf{w}_1,\ldots,\mathbf{w}_{n-1},\mathbf{w}_{n+1},\ldots,\mathbf{w}_N)$ is obtained. The output of *sliding sweep* algorithm is the required APP of the codeword symbol. The decision about the actual value of the transmitted symbol is made based on LLR's sign.

The values obtained by the *sliding-sweep* method are applied to (4), and the decision based on a logarithm sign is made:

$$\operatorname{sgn}(\Lambda_n(\alpha)) = \operatorname{sgn}\left(\log \frac{P(c_n = \alpha \mid \mathbf{r}, \alpha \neq 0)}{P(c_n = 0 \mid \mathbf{r})} \right) = \begin{cases} \alpha, if \ \Lambda > 0, \\ 0, \text{otherwise.} \end{cases} \tag{10}$$

Table 1 illustrates the results of the *sliding-sweep* algorithm applied to the Hamming code (K,N)=(4,7) [1].

Table 1

Outputs of the sliding sweep algorithm

r	0_1	0_2	0_1	0_2	1_1	0_2	0_2
$P(\mathbf{r}_n^{\perp} \mid c_n = 1)$	1,08e–3	1,13e–3	1,09e–3	2,04e–3	0,82e–3	2,04e–3	1,4e–3
$P(c_n = 0 \mid \mathbf{r})$	3,26e–4	1,07e–4	3,26e–4	1,02e–4	1,75e–3	1,02e–4	1,05e–4
$\Lambda_n(1)$	–0,168	–0,724	–0,168	–1	1,153	–1	–0,823

Analyzing $P(c_n= 0|\mathbf{r})$ and $P\left(\mathbf{r}_n^{\perp} \mid c_n = 1\right)$ values of the received vector $r=\{0_1,0_2,0_1,0_2,1_1,0_2,0_2\}$, we can conclude that the error is present in r_5, but using Λ logarithm sign, the decision about the error in the specified bit can be made.

Sliding sweep algorithm:
input: $P(r \mid v)$ – channel statistics; $\mathbf{W}_1,\ldots,\mathbf{W}_N$.
for n **in** 1 **to** N **do**
1. **for** α **in** 0 **to** 2^m-1 **do**
2. **generate :**
$\mathbf{Y}(n)= [y_i]_{Q\times1}$, **where** y_i=1 **if** $\forall j \exists i$ (w_{ij}= α) & y_i=0 **else**;
$\mathbf{X}(n)= \langle x_j \rangle_{1\times Q}$, **where** x_j=1 **if** $\forall j \exists i$ (w_{ij}= α) & x_j=0 **else**;
3. **for** k **in** n-1 **downto** 1 **do** -- backward
$\mathbf{W}_k := \mathbf{W}_k$ **mask** $\mathbf{Y}_k$; $\mathbf{Y}_{k-1} := \mathbf{W}_k \otimes \mathbf{Y}_k^T$;
4. **for** k **in** n+1 **to** N **do** -- forward
$\mathbf{W}_k := \mathbf{W}_k$ **mask** $\mathbf{X}_k$; $\mathbf{X}_{k+1} := \mathbf{X}_k \otimes \mathbf{W}_k$
5. **for** k **in** 1 **to** N **do** $\mathbf{W}_k \rightarrow \mathbf{M}_k$; - - map

output: $P\left(\mathbf{r}_n^{\perp} \mid c_n = \alpha\right) = \mathbf{M}_0 \left(\prod_{\substack{k=1 \\ k\neq n}}^{N} \mathbf{M}_k \right) \mathbf{M}_0^T$,

where $\mathbf{M}_0$=<1 0 … 0>.

Fig. 2. Sliding-sweep algorithm

Conclusion

The backward algorithm generally allows to generate the syndrome trellis for (non)binary linear block code. The elaborated sliding-sweep procedure based on forward and backward APP-calculus, in comparison with one-sweep algorithm [2], estimates the LLR, which makes it possible to apply the corrections based on the decision that finally permits to select more reliable symbol.

REFERENCES

1. Johansson T., Zigangirov K. A simple one-sweep algorithm for optimal APP symbol decoding of linear block codes // IEEE Trans. Inform. Theory.— 1998.— Vol. 44.

2. Johansson T., Trofimov A. One-sweep APP decoding algorithm for binary block codes with reduced trellis memory // Proceedings of International workshop “Coding Theory Days in St. Petersburg”.— 2008.— P. 88—93.

3. Pocotilenco V. Soft decoding of data received from nano-satellite “SATUM” // Proceedings of ICTEI conference.— Moldova, Chisinau.— 2012.— P. 252—255.

4. Hagenauer J., Offer E., Papke L. Iterative decoding of binary block and convolutional codes // IEEE Trans. Inform. Theory.— 1996.— Vol. 42.— P. 429—445.

В. В. Покотиленко, Г. К. Бодян , Т. В. Шестакова

Улучшенный метод мягкого декодирования данных, переданных по каналу связи.

Искажения сигнала при передаче данных являются проблемой, решаемой на разных фазах существования передаваемых данных, начиная с их генерации и собственно передачи по каналу связи и кончая их обработкой при приеме и после. В данной работе предложен метод *sliding-sweep* для мягкого декодирования линейных недвоичных кодов, который упрощает вычисление апостериорных значений вероятностей. При этом предлагается принятие решения о наличии ошибки выполнять на основе апостериорных оценок правдоподобия.

Ключевые слова: *прием и обработка сигналов, треллис, мягкое декодирование.*