PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

Implementation complexity analysis of the turbo decoding algorithms on digital signal processor

Olexander Romanyuk, Yuriy Ivanov, Oleg Bisikalo, Oleg Stukach, Ruslana Ignatovska, et al.

Olexander N. Romanyuk, Yuriy Yu. Ivanov, Oleg V. Bisikalo, Oleg V. Stukach, Ruslana V. Ignatovska, Ryszard S. Romaniuk, Zhanar Azeshova, "Implementation complexity analysis of the turbo decoding algorithms on digital signal processor," Proc. SPIE 10808, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2018, 1080820 (1 October 2018); doi: 10.1117/12.2501504

Event: Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2018, 2018, Wilga, Poland

Implementation complexity analysis of the turbo decoding algorithms on digital signal processor

Olexander N. Romanyuk*[a], Yuriy Yu. Ivanov[a], Oleg V. Bisikalo[a], Oleg V. Stukach[b], Ruslana V. Ignatovska[c], Ryszard S. Romaniuk[d], Zhanar Azeshova[e]
[a]Vinnytsia National Technical University, Vinnytsia, Ukraine; [b]National Research Tomsk Polytechnic University, Tomsk, Ukraine; [c]Odessa State University of Internal Affairs, Odessa, Ukraine; [d]Warsaw University of Technology, Warsaw, Poland; [e]Kazakh National Research Technical University named after K.I.Satpayev, Almaty, Kazakhstan

ABSTRACT

In this article, we work with turbo codes, which are widely used to reduce bit error rate in digital communication systems. we start with a brief discussion of the mathematical apparatus, which is connected with the turbo decoding algorithms. the key implementation issue for these algorithms is the overall high decoding complexity. therefore have been presented some estimations of the implementation complexity of the turbo decoding algorithms on digital signal processor.

Keywords: error-correcting coding, turbo code, iterative turbo decoding, MAP, SOVA, number of math elementary operations, digital signal processor

1. INTRODUCTION

Channel coding or forward error correction (FEC), introduces redundancy into a data signal, so that errors incurred during transmission can be corrected at the receiver. Succeeding decades saw the proposal of many FEC coding schemes in pursuit of this purpose. Well-known examples include algebraic, convolutional, product, concatenated, low-density parity-check codes, etc[1,2]. One of the most promising direction in the development of an error correction theory is data protection, which is based on parallel concatenation of convolutional codes or turbo codes. It is shown that at the high bit error rate (BER = 10^{-5}-10^{-7} from 0 to 3 dB) the best of the considered codes is turbo code (power gain of about 7-9 dB), what is very close to Shannon limit[3]. Thus, turbo code is the most effective method for information transmission in channels with low energy consumption in many communication systems (television, telemetry, wireless local area networks, software defined radio systems, mobile and space communication, etc).

So now scientists perform detailed researches in the turbo decoding field with using of complex computational iterative decoding algorithms with high error correction degree. There are two main turbo decoding algorithms, such as soft-output Viterbi algorithm (SOVA) and BCJR or MAP (maximum a posteriori) algorithm[4,5]. The SOVA differs from the classical Viterbi algorithm, in that it uses a modified path metric, which takes into account the a priori probabilities of the input symbols, and produces a soft output indicating the reliability of the decision. Bi-SOVA algorithm works as bidirectional SOVA. The MAP is symbol-by-symbol decoding algorithm with high decoding complexity. In log-MAP turbo decoding, the complicated log exponential sum (in MAP algorithm) is often simplified with the Jacobian logarithm which consists of the max operation along with some correction function. Although the max-log-MAP algorithm reduces the complexity of the Jacobian logarithm implementation by omitting the compensation function. All turbo decoding algorithms can be interpreted as the implementation of techniques known as message passing algorithm or belief propagation algorithm. To describe the strategy, that is embodied in an iterative decoding of turbo codes, Hagenauer introduced the concept of "turbo"-principle[6-8].

Turbo codes can be implemented in software, hardware or mixed (software and hardware) manner on digital signal processors (DSP)[7-10]. However, turbo codes require decoders with large computational complexity, because the number of mathematical operations required by turbo decoders increases exponentially with the increase of constraint length of constituent convolutional codes. So the *purpose* of this work is analytically determine functional dependencies and provide the comparative analysis for implementation complexity of turbo decoding algorithms on DSP.

* rom8591@gmail.com

Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2018, edited by Ryszard S. Romaniuk, Maciej Linczuk, Proc. of SPIE Vol. 10808, 1080820

The paper is structured as follows. Section II is connected with the brief overview of turbo decoding algorithms. In Section III we calculate the total number of elementary math operations for turbo decoding algorithms, which are doing on DSP. Then we provide the comparative complexity analysis between popular turbo decoding algorithms. The paper is finally concluded in Section IV.

2. MATH OF THE TURBO DECODING ALGORITHMS

The math basis of the maximum a posteriori turbo decoding algorithm is connected with computational procedure for estimation the solutions reliability or logarithm of the likelihood ratio (LLR)[3, 11]. The LLRs can be written in the formula

$$LLR_{MAP}(d_k) = \ln \frac{p(d_k = +1 \mid x)}{p(d_k = -1 \mid x)} = \ln \frac{\sum\limits_{d_k=+1}^{(s',s)} p(s',s,x)}{\sum\limits_{d_k=-1}^{(s',s)} p(s',s,x)} \mathop{\gtrless}_{H_2}^{H_1} 0, \tag{1}$$

where d_k – information symbol at the time k on trellis; $x = \{x_k\}$ – received sequence; s' and s – past and present states at trellis; H_1, H_2 is the hypotheses and if $LLR_{MAP}(d_k) > 0$ (H_1 is true), the hard binary decision will be 1, in otherwise – 0.

After simplifications the generalized expression for the LLR computation procedure for MAP decoding algorithm has the form[11-13]

$$LLR_{MAP}(d_k) = LLR_{ch} \cdot x_k + LLR_{apr}(d_k) + LLR_{ext}(d_k) = \ln \frac{\sum\limits_{d_k=+1}^{(s',s)} \alpha_{k-1}(s') \cdot \gamma_k(s',s) \cdot \beta_k(s)}{\sum\limits_{d_k=-1}^{(s',s)} \alpha_{k-1}(s') \cdot \gamma_k(s',s) \cdot \beta_k(s)}, \tag{2}$$

where $LLR_{ch}, LLR_{apr}, LLR_{ext}$ – channel, a priori and extrinsic log-likelihood ratios for turbo decoding process; ln(.) – natural logarithm; $\alpha_k(s) = \sum\limits_{(s',s)} \gamma_k(s',s) \cdot \alpha_{k-1}(s')$ – forward path metric on trellis for recursive systematic convolutional (RSC) code; $\beta_{k-1}(s') = \sum\limits_{(s',s)} \gamma_k(s',s) \cdot \beta_k(s)$ – backward path metric on the trellis for RSC code; $\gamma_k(s',s)$ – transit or rib metric, which is calculated as[12]

$$\gamma_k(s',s) \approx \exp\left(\frac{1}{2} \cdot (d_k \cdot LLR_{apr}(d_k)) + d_k \cdot LLR_{ch.} \cdot x_k + \left(\sum_{k=2}^{n} d_{k,v} \cdot LLR_{ch.} \cdot x_{k,v} \right) \right), \tag{3}$$

where $d_{k,v}$, $x_{k,v}$ – parity symbol from RSC encoder and its noisy version.

The logarithmic metrics are used in log-MAP algorithm for reducing the computational complexity. The metrics are presented in the next form[15]

$$\alpha_k^{LM}(s) = \ln \alpha_k(s) = \ln(\sum_{(s',s)} \exp(\gamma_k^{LM}(s',s) + \alpha_{k-1}^{LM}(s'))), \tag{4}$$

$$\beta_{k-1}^{LM}(s') = \ln \beta_{k-1}(s') = \ln(\sum_{(s',s)} \exp(\gamma_k^{LM}(s',s) + \beta_k^{LM}(s))), \tag{5}$$

$$\gamma_k^{LM}(s',s) = \ln \gamma_k(s',s) = \frac{1}{2} \cdot \left(d_k \cdot LLR_{apr.}(d_k) + d_k \cdot LLR_{ch.} \cdot x_k + \left(\sum_{k=2}^{n} d_{k,v} \cdot LLR_{ch.} \cdot x_{k,v} \right) \right). \tag{6}$$

The function of the sum of exponential components is presented recursively in the formula

$$f(A_1...A_N) = \ln\sum_{i=1}^{N} e^{A_i} = f(A_1, f(A_2,..., f(A_{N-2}, f(A_{N-1}, A_N)))) , \tag{7}$$

where simplification can be represented in the form with the correction function f_{cor}

$$\begin{aligned} f(A_{N-1}, A_N) &= \ln(\exp(A_{N-1}) + \exp(A_N)) = \\ &= \max(A_{N-1}, A_N) + \ln(1+\exp(-\mid A_{N-1} - A_N \mid)) = \max(A_{N-1}, A_N) + f_{cor}(z). \end{aligned} \tag{8}$$

So the generalized expression for the *LLR* computation procedure for log-MAP decoding algorithm can be presented as

$$\begin{aligned} LLR_{LM}(d_k) &= \ln\frac{\sum\limits_{d_k=+1}^{(s',s)} \exp(\alpha_{k-1}^{LM}(s'))\cdot\exp(\beta_k^{LM}(s))\cdot\exp(\gamma_k^{LM}(s',s))}{\sum\limits_{d_k=-1}^{(s',s)} \exp(\alpha_{k-1}^{LM}(s'))\cdot\exp(\beta_k^{LM}(s))\cdot\exp(\gamma_k^{LM}(s',s))} = \\ &= LLR_{ch}\cdot x_k + LLR_{apr}^{LM}(d_k) + \ln\frac{\sum\limits_{d_k=+1}^{(s',s)} \exp(\alpha_{k-1}^{LM}(s') + \beta_k^{LM}(s) + \gamma_k^{ext}(s',s))}{\sum\limits_{d_k=-1}^{(s',s)} \exp(\alpha_{k-1}^{LM}(s') + \beta_k^{LM}(s) + \gamma_k^{ext}(s',s))}. \end{aligned} \tag{9}$$

The manner in which the correction function is calculated is critical to the performance and complexity of the decoding. Several methods have been proposed to simplify its computation, which gives a tradeoff between complexity and performance, but such decoding algorithms are suboptimal. For example, the formula for calculation compensation term for piecewise linear PL-log-MAP algorithm is presented in the next form

$$f_{cor}(z) = \begin{cases} -0{,}3792\cdot z + 0{,}6754, & if \quad z\in[0;1); \\ -0{,}2229\cdot z + 0{,}5327, & if \quad z\in[1;1{,}5); \\ -0{,}1483\cdot z + 0{,}4213, & if \quad z\in[1{,}5;2); \\ -0{,}0773\cdot z + 0{,}2758, & if \quad z\in[2;3); \\ -0{,}0300\cdot z + 0{,}1362, & if \quad z\in[3;4]; \\ +0{,}0100, & if \quad z\in(4;\infty). \end{cases} \tag{10}$$

This approximation is more accurate than other compensation functions.
Another simple algorithm for turbo decoding is max-log-MAP, the forward, backward metrics and *LLR* of which can be calculated without correction term in the form

$$LLR_{MLM}(d_k) = \max_{d_k=+1}^{(s',s)}(\alpha_{k-1}^{MLM}(s') + \beta_k^{MLM}(s) + \gamma_k^{LM}(s',s)) - \max_{d_k=-1}^{(s',s)}(\alpha_{k-1}^{MLM}(s') + \beta_k^{MLM}(s) + \gamma_k^{LM}(s',s)). \tag{11}$$

Another iterative decoding algorithm is SOVA[16,17], which uses metrics on the trellis and add-compare-select operation block. The calculation of the forward path metric is presented in the formula

$$M_k(s_k) = \max_{d_k=\pm1}^{(s',s)}(M_{k-1}(s_{k-1}) + \gamma^{LM}{}_k(s',s)) , \tag{12}$$

where $M_{k\text{-}1}(s_{k\text{-}1})$ – current metric on trellis.

The symbol-by-symbol SOVA method is evaluate the reliability of the binary symbol based on Hagenauer's formula[17] on the obtained maximally likelihood path on trellis with a sliding window of size δ.

$$LLR_{SOVA} \approx d_k \cdot \min_{\substack{i=k...k+\delta \\ d_k \neq d_k^i}} (|M_k^{d_k=+1}(s_k) - M_k^{d_k=-1}(s_k)|) = d_k \cdot \min_{\substack{i=k...k+\delta \\ d_k \neq d_k^i}} \Delta_i^{s_i} . \tag{13}$$

The Bi-SOVA algorithm[18] is performed forward and backward by formulas (12) and (13). A posteriori decisions are calculated in the next form

$$LLR_{Bi-SOVA}(d_k) = \begin{cases} LLR_{SOVA}^{\rightarrow}(d_k), \ if \ LLR_{SOVA}^{\rightarrow}(d_k) > LLR_{SOVA}^{\leftarrow}(d_k); \\ LLR_{SOVA}^{\leftarrow}(d_k), \ if \ LLR_{SOVA}^{\leftarrow}(d_k) > LLR_{SOVA}^{\rightarrow}(d_k). \end{cases} \tag{14}$$

where $LLR_{SOVA}^{\rightarrow}(d_k)$, $LLR_{SOVA}^{\leftarrow}(d_k)$ – a posteriori decisions on the forward and backward SOVA algorithm. You can also perform the normalization of metrics to reduce the numerical barrier.

3. COMPARATIVE COMPLEXITY ANALYSIS OF TURBO DECODING ALGORITHMS ON DSP

Any RSC encoder is described by trellis, in which the number of possible states is $T = 2^m$, then the total number of transitions is $L = 2T = 2^{m+1} = 2K$, where K is the constraint length of the encoder, m is the encoder memory. The division operation is presented as $a \cdot (1 / b)$, the exponent function exp(.) we expand to series in the range ($-\infty$; $+\infty$), the logarithm ln(.) we present as a series in the range (0; $+\infty$). Note that d^{hard} = sign(LLR_{apost}), and the value of $n(\delta)$ is the number of "min" operations n on the sliding window of size δ (let $n(\delta) = 5K - 1 = 5m + 4$). Consequently, all operations will be elementary math operations (EMO) for the DSP[19,20]. Thus, taking into account the above, we will show in tables 1-5 (where q is the total number of symbols from RSC encoder; symbols with $\widetilde{x}$ are normalized) analytic expressions for determining the number of EMO for decoding the information bit on the DSP using different turbo decoding algorithms.

Table 1. The number of EMO for decoding the information bit on DSP using MAP algorithm.

EMO	Parameters of MAP turbo decoding algorithm							
	$\gamma(s',s)$	$\alpha_k(s)$	$\beta_{k-1}(s')$	$\widetilde{\alpha}_k(s)$	$\widetilde{\beta}_{k-1}(s')$	LLR_{MAP}	LLR_{MAP}^{ext}	d_k^{hard}
ADD	$2^{m+1} \cdot q + 11 \cdot 2^{m+1}$	2^m	2^m	$2^m - 1$	$2^m - 1$	$2^{m+1} + 21$	-	-
SUB	-	-	-	-	-	12	2	-
MULT	$2^{m+1} \cdot q + 123 \cdot 2^{m+1}$	2^{m+1}	2^{m+1}	2^m	2^m	$2 \cdot 2^{m+1} + 311$	1	1
RECIPS	$10 \cdot 2^{m+1}$	-	-	2^m	2^m	13	-	1
ABS	-	-	-	-	-	-	-	1

Table 2. The number of EMO for decoding the information bit on DSP using PL-log-MAP algorithm.

EMO	Parameters of PL-log-MAP turbo decoding algorithm							
	$\gamma(s',s)$	$\alpha_k(s)$	$\beta_{k-1}(s')$	$\tilde{\alpha}_k(s)$	$\tilde{\beta}_{k-1}(s')$	$LLR_{PL-\log-MAP}$	$LLR_{PL-\log-MAP}^{ext}$	d_k^{hard}
ADD	$2^{m+1}\cdot q$	2^{m+2}	2^{m+2}	-	-	$2^{m+3}-4$	-	-
SUB	-	2^m	2^m	2^m	2^m	$2^{m+1}-1$	2	-
MULT	$2^{m+1}\cdot(q+3)$	2^m	2^m	-	-	$2^{m+1}-2$	1	1
RECIPS	-	-	-	-	-	-	-	1
MAX	-	2^m	2^m	2^m-1	2^m-1	$2^{m+1}-2$	-	-
COMP	-	$3\cdot 2^{m+1}$	$3\cdot 2^{m+1}$	-	-	$12\cdot 2^m-12$	-	-
ABS	-	2^m	2^m	-	-	$2^{m+1}-2$	-	1

Table 3. The number of EMO for decoding the information bit on DSP using max-log-MAP algorithm.

EMO	Parameters of max-log-MAP turbo decoding algorithm							
	$\gamma(s',s)$	$\alpha_k(s)$	$\beta_{k-1}(s')$	$\tilde{\alpha}_k(s)$	$\tilde{\beta}_{k-1}(s')$	LLR_{MLM}	LLR_{MLM}^{ext}	d_k^{hard}
ADD	$2^{m+1}\cdot q$	2^{m+1}	2^{m+1}	-	-	2^{m+2}	-	-
SUB	-	-	-	2^m	2^m	1	2	-
MULT	$2^{m+1}\cdot(q+3)$	-	-	-	-	-	1	1
RECIPS	-	-	-	-	-	-	-	1
MAX	-	2^m	2^m	2^m-1	2^m-1	$2^{m+1}-2$	-	-
ABS	-	-	-	-	-	-	-	1

Table 4. The number of EMO for decoding the information bit on DSP using SOVA algorithm.

EMO	Parameters of SOVA turbo decoding algorithm						
	$\gamma(s',s)$	$M_k(s_k)$	$\tilde{M}_k(s_k)$	$\Delta_i^{s_i}$	LLR_{SOVA}	LLR_{SOVA}^{ext}	d_k^{hard}
ADD	$2^{m+1}\cdot q$	2^{m+1}	-	-	-	-	-
SUB	-	-	2^m	1	-	2	-
MULT	$2^{m+1}\cdot(q+3)$	-	-	-	1	1	1
RECIPS	-	-	-	-	-	-	1
MAX	-	2^m	2^m-1	-	-	-	-
MIN	-	-	-	-	$5\cdot m+4$	-	-
ABS	-	-	-	-	-	-	1

Table 5. The number of EMO for decoding the information bit on DSP using Bi-SOVA algorithm.

EMO	**Parameters of Bi-SOVA turbo decoding algorithm**						
	$\gamma(s',s)$	$M_k(s_k)$	$\widetilde{M}_k(s_k)$	$\Delta_i^{s_i}$	$LLR_{Bi-SOVA}$	$LLR_{Bi-SOVA}^{ext}$	d_k^{hard}
ADD	$2^{m+1}\cdot q$	2^{m+2}	-	-	-	-	-
SUB	-	-	2^{m+1}	2	-	2	-
MULT	$2^{m+1}\cdot(q+3)$	-	-	-	2	1	1
RECIPS	-	-	-	-	-	-	1
MAX	-	2^{m+1}	$2^{m+1}-2$	-	-	-	-
MIN	-	-	-	-	$2\cdot(5\cdot m+4)$	-	-
COMP	-	-	-	-	1	-	-
ABS	-	-	-	-	-	-	1

By adding all operations, we obtain the total number of EMO for each of the considered turbo decoding algorithms. Consequently, the complexity of decoding the information bit for the corresponding algorithms is represented by the function $f(m,\ q)$, the maximum of which in (+ ∞; + ∞), and the minimum at the point [1; 2]. The total number of elementary mathematical operations that must be performed by the DSP for each of the turbo decoding algorithms is determined by the following functions

$$f_{MAP}(m,q)=2^{m+2}\cdot q+153\cdot 2^{m+1}+361, \tag{15}$$

$$f_{PL-\log-MAP}(m,q)=2^{m+2}\cdot q+33\cdot 2^{m+1}-19, \tag{16}$$

$$f_{\max-\log-MAP}(m,q)=2^{m+2}\cdot q+11\cdot 2^{m+1}+3, \tag{17}$$

$$f_{SOVA}(m,q)=2^{m+2}\cdot q+11\cdot 2^{m}+5\cdot m+11, \tag{18}$$

$$f_{Bi-SOVA}(m,q)=2^{m+2}\cdot q+2^{m+4}+10\cdot m+17\,. \tag{19}$$

The analysis shows that the implementation complexity of turbo decoding algorithms increases in the form of the power function from m, and with increasing q the function increases by 2^{m+2}. For the comparative analysis of the computational complexity of the turbo decoding algorithms, the relative parameters $O = N/n$ are used, where N and n – respectively the larger and smaller numbers of the mathematical operations for the decoding algorithms, that are compared. The results of calculations for m (from 1 to 9) and q (from 2 to 6) values are presented in the Fig. 1.

In addition, with the same output, the most complicated is the MAP algorithm, followed by PL-log-MAP, max-log-MAP, Bi-SOVA, and the least complicated is SOVA. The obtained results can be used to analyze the implementation complexity of turbo decoder on the DSP in order to select an element base.

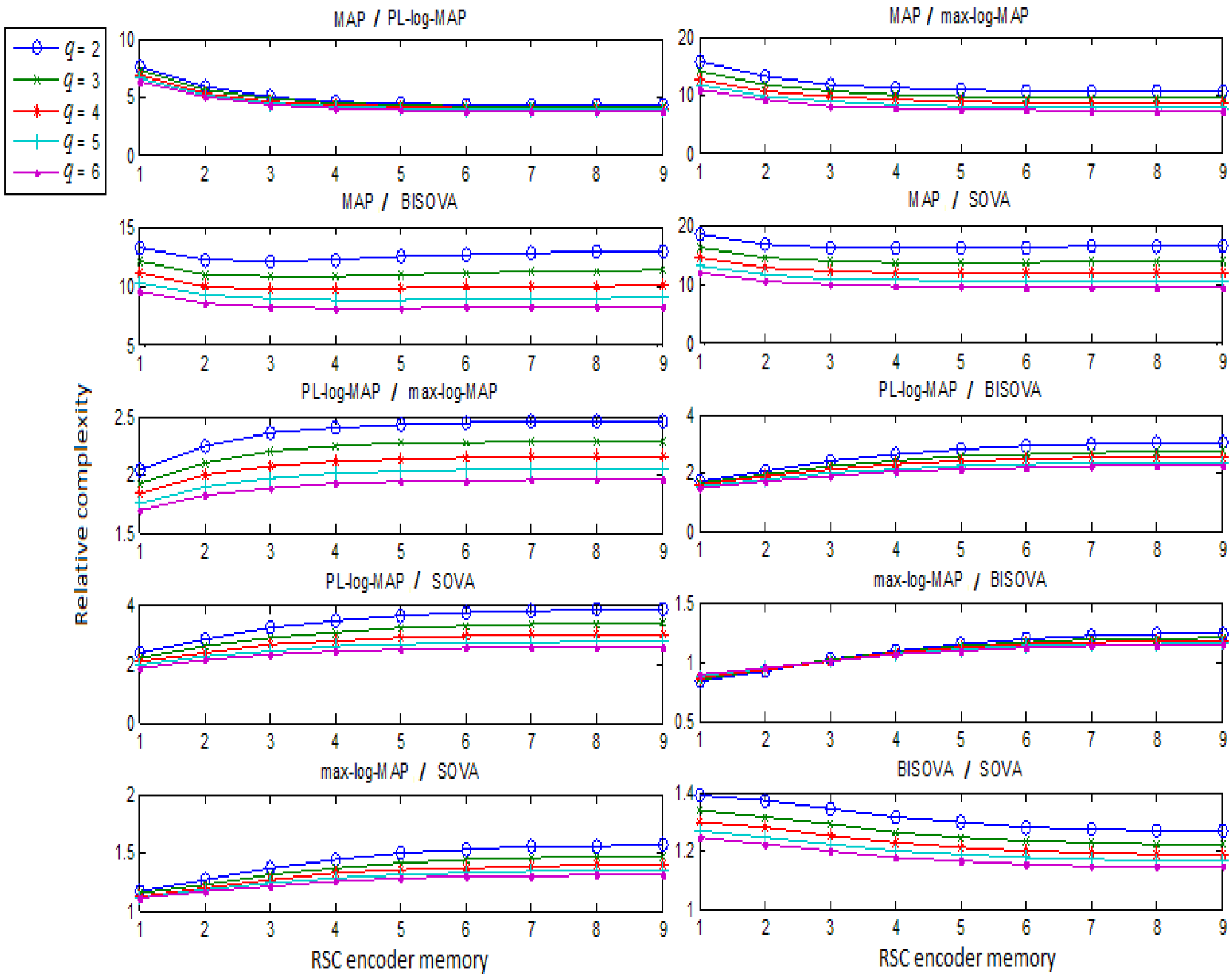

Figure 1. Relative complexity of implementation for various turbo decoding algorithms on DSP

4. CONCLUSIONS

Thus, given the complexity of the task of decoding and implementing the turbo decoder, have been determined some features that lead to its solution. In this paper, we describe the math of various turbo decoding algorithms. Also we deduce and consider the functional dependencies $f(m, q)$, that determine the number of EMO (taking into account the number of memory cells and the total number of symbols from the RSC encoder output) for various turbo decoding algorithms. In order to compare the complexity of the implementation of the turbo decoding algorithms, EMO, which are necessary for decoding the information bit, should be presented as elementary for DSP.

Analytical expressions for assessing the complexity of the turbo decoding algorithms can be used for further analysis of the complexity of turbo decoder hardware-software implementation in digital communication systems of different functional purposes.

REFERENCES

[1] Declerq, D., Fossorier, M., Biglieri, E. et al., "Channel Coding: Theory, Algorithms, and Applications," Academic Press Library in Mobile and Wireless Communications, 2-52 (2014).

[2] Berrou, C., Douillard C., Jezequel, M. et al., "Channel Coding in Communication Networks. From Theory to Turbocodes," Chippenham, 255-371 (2007).

[3] Berrou, C., Glavieux, A., and Thitimajshima, P., "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes," Proceedings of ICC'93, Geneva, Switzerland, 1064-1070 (1993).

[4] Moon, T. K., [Error Correcting Coding: Mathematical Methods and Algorithms], John Wiley & Sons, New Jersey, 581-633 (2005).

[5] Kvyetnyy, R. N., Ivanov, Yu. Yu., Krivogubchenko, S.G., and Stukach, O.V., "Features of Estimation the Transmission Data Process Using Turbo-Codes," Journal Metrology and Devices 3, 25-32 (2017).

[6] Hagenauer, J., "The Turbo Principle: Tutorial Introduction and State of the Art," Proclaimed on the International Symposium on Turbo Codes and Related Topics, 1-11 (1997).

[7] Pietrobon, S. S., "Implementation and Performance of a Turbo/MAP Decoder," International Journal of Satellite Communications 16(1), 23-46 (1998).

[8] Wu, Y., [Implementation of Parallel and Serial Concatenated Convolutional Codes - Ph.D. dissertation], Faculty of the Virginia Polytechnic Institute and State University, Blacksburg (2000).

[9] Kotyra, A., Wojcik, W., Gromaszek, K. et al. "Assessment of biomass-coal co-combustion on the basis of flame image," Przegląd Elektrotechniczny 88(11B), 295-297 (2012).

[10] Olchowy, D., "Electrical impedance tomography hardware system based on low power digital signal processor," Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska – IAPGOŚ 3, 57-58 (2013).

[11] Abrantes, S., [From BCJR to Turbo Decoding: MAP Algorithms Made Easier], Information and Telecommunication Technology Center of the University of Kansas, Lawrence (1996).

[12] Soleymani, M. R., Gao, Y., Vilaipornsawai U., [Turbo Coding for Satellite and Wireless Communications], Kluwer Academic, New York, 23-52 (2002).

[13] Hanzo, L., Liew, T. H., and Yeap, B. L., [Turbo Coding, Turbo Equalisation and Space-Time Coding for Transmission over Wireless Channels], Department of Electronics and Computer Science of UK, Southampton, 107-170 (2002).

[14] Jiang, Y., [A Practical Guide to Error-Control Coding Using MATLAB], Artech House, UK, 213-251 (2010).

[15] Robertson, P., Villebrun, E., and Hoeher, P., "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," IEEE International Conference on Communications ICC'95 "Gateway to Globalization" 2, 1009-1013 (1995).

[16] Hagenauer, J., and Hoeher P., "A Viterbi Algorithm with Soft-Decision Outputs and its Applications," Proceedings of IEEE Global Telecommunications Conference, 1680-1686 (1989).

[17] Woodard, J., and Hanzo, L., "Comparative Study of Turbo Decoding Techniques: An Overview," IEEE Transactions on Vehicular Technology 6(49), 2208-2233 (2000).

[18] Chen, J., Fossorier, M. P. C., Lin, S., and Xu, C. "Bi-directional SOVA Decoding for Turbo-codes," IEEE Communications Letters 4(12), 405-407 (2000).

[19] Malardel, F., [Simulation and Optimisation of the Turbo Decoding Algorithm - Ph.D. dissertation], Ecole Francaise d'Electronique et d'Informatique, France (1996).

[20] Bhise, A., Vyavahare, P. "Complexity Analysis of Iterative Decoders in Mobile Communication Systems," International Journal of Information and Electronics Engineering 4(2), 121-128 (2014).