

## ОСОБЛИВОСТІ АПАРАТНО-ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ТУРБО-КОДІВ: АНАЛІЗ СКЛАДНОСТІ РЕАЛІЗАЦІЇ НА ЦИФРОВОМУ СИГНАЛЬНОМУ ПРОЦЕСОРІ

<sup>1</sup>Вінницький національний технічний університет

Проведено огляд алгоритмів, які використовуються для декодування турбо-кодів. Оцінено складність реалізації декодування біта інформації у ході апаратно-програмної реалізації турбо-кода. Встановлено та проаналізовано функціональну залежність різних алгоритмів декодування турбо-кодів від пам'яті та загальної кількості бітів з кодерів.

**Ключові слова:** завадостійке кодування, турбо-код, алгоритми декодування, MAP, log-MAP, PL-log-MAP, max-log-MAP, Bi-SOVA, SOVA, кількість елементарних математичних операцій, цифрові сигнальні процесори.

### Вступ

Потужним кроком в напрямі до ідей С.Е. Shannon є турбо-код (ТК) або паралельна конкатенація згорткових кодерів (*PCCC* — Parallel Concatenated Convolutional Codes), яку вперше запропонували у своїй фундаментальній роботі С. Berrou, А. Glavieux, Р. Thitimajshima, підсумувавши роботу, яку вели математики, а також фахівці з апаратних, програмних засобів EOM та проблем передавання інформації, розробивши технологію, яка дозволяла досягти рекордного *BER* (Bit Error Rate — коефіцієнт бітових помилок) за низьких відношень сигнал/шум [1]. Ідея їхньої роботи полягала у вдосконаленні алгоритму, який вперше оприлюднили L. Bahl, J. Cocke, F. Jelinek, J. Raviv [2]. Робота ТК заснована на таких принципах: подача вихідних даних назад на вхід, який аналогічний турбонаддуву двигуна; застосована концепція ітеративних імовірнісних декодерів з м'яким входом та м'яким виходом (*SISO* — Soft-Input Soft-Output); використання паралельно працюючих елементарних кодерів [3].

Підвищення завадостійкості є важливим завданням під час проектування сучасних завадостійких систем передавання цифрової інформації. Швидкість передавання даних каналом зв'язку безпосередньо пов'язана з якістю і швидкістю декодування. Декодери завадостійких кодів є найскладнішими для апаратної та програмної реалізації, оскільки під час проектування сучасних інформаційних систем до них висуваються умови роботи з необхідною швидкістю передавання інформації відповідно до характеристик каналу зв'язку. Для дослідження роботи цифрових систем зв'язку різного функціонального призначення необхідними етапами є: 1) аналіз складності декодера; 2) розробка апаратної архітектури та програмного забезпечення з використанням мов програмування високого рівня; 3) проведення комп'ютерного ймовірнісного та математичного моделювання характеристик коду.

У цій роботі розв'язана перша задача, пов'язана зі складністю процесу декодування в ході роботи ітеративного ТК декодера. Висока ефективність ТК зумовлена розробленими для них спеціальними ітеративними алгоритмами декодування *SOVA* [4—6], *Bi-SOVA* [7, 8], *MAP* [2, 6, 7, 9—12], *log-MAP* [6—7, 9, 11—12] та іншими модифікаціями [9, 11, 13, 14]. Порівняльний аналіз складності реалізації декодування біта інформації для алгоритмів декодування ТК, наведений в роботах [15—17]. Зазначимо, що дослідження цього питання з використанням цифрових сигнальних процесорів (*DSP* — Digital Signal Processor) розпочато у роботі [16], але в ній не розглянуто розв'язання цієї задачі для алгоритмів декодування *SOVA* [4—6], *Bi-SOVA* [7, 8] та *PL-log-MAP* [14], крім того враховані не всі елементарні операції, які виникають в процесі розрахунку апостеріорних значень *LLR* (Log-Likelihood Ratios — логарифм відношень функцій правдоподібності). Тому виникає задача визначення складності реалізації декодування біта інформації в кількості елементарних математичних операцій (ЕМО), необхідних для декодування одного біта інформації, з урахуванням перерахованих особливостей.

ТК мають високі коректувальні властивості за низьких відношень сигнал/шум в каналі зв'язку і можуть бути реалізовані програмним, апаратним або програмно-апаратним способом. Як правило,

практично ТК реалізуються програмно-апаратним способом на цифрових сигнальних процесорах, наприклад, на високопродуктивному 32-розрядному процесорі ADSP-2106x сімейства ADSP-21000 з плаваючою точкою, який використовується для обробки різноманітних мультимедіа даних. Процесори сімейства ADSP-21000 виконують всі команди за один цикл, підтримуючи високу тактову частоту, а також застосовують повний набір арифметичних операцій, який включає крім додавання (ADD), віднімання (SUB), множення (MULT) ще й примітиви ділення  $1/b$  (RECIPS), квадратний корінь (SQRT(10)), порівняння (COMPARE), визначення абсолютного значення (ABS), операції мінімуму (MIN) та максимуму (MAX), арифметичний (SHIFT) та циклічний (ROTATE) зсув тощо. Всі обчислювальні операції виконуються в арифметико-логічному пристрої, помножувачі і елементі зсуву обчислювального пристрою *DSP* [16].

Метою роботи є визначення кількості ЕМО для здійснення порівняльного аналізу складності реалізації декодування біта інформації на *DSP* за використання різних алгоритмів декодування ТК з урахуванням кількості комірок пам'яті  $m \in [1; \infty)$  для *RSC* (Recursive Systematic Convolutional — рекурсивного систематичного згорткового) кодера і загальної кількості символів  $q \in [2; \infty)$  з виходу *RSC* кодера, що дозволить виконати вибір елементної бази для побудови ТК.

### Алгоритм *MAP* та його модифікації *PL-log-MAP*, *max-log-MAP*

Під час роботи турбо-кодера в канал зв'язку передаються два кодові блоки: перший блок, який складається з інформаційної частини  $X = d_k$  і перевіркової частини  $d_{k,2}$  кодера 1, і другий кодовий блок, що складається з перемішаної інформаційної частини  $\pi(d_k)$  і перевіркової частини  $d_{k,3}$  з кодера 2. Під час декодування турбо-декодером із загального потоку даних виділяють два зашумлені кодові блоки  $(x_k, x_{k,2})$  та  $(x_k, x_{k,3})$ , причому інформаційні частини цих двох блоків через систематичне кодування і з урахуванням перемішування ідентичні. Оскільки інформаційні частини кожного з двох кодових блоків ідентичні, декодовану інформацію першого (другого) декодера  $LLR_{ext.}(d_k)$  (зовнішня інформація) з урахуванням перемішування  $\pi$  можна використовувати як апіорну інформацію  $LLR_{apr.}(d_k)$  для другого (першого) декодера з метою уточнення результату декодування, тобто інформація обертається петлею до того моменту, поки м'які рішення не сходяться на стабільному наборі значень. Крім того, для уточнення апостеріорного  $LLR(d_k)$  використовується каналне  $LLR_{ch.}$ . Далі опишемо в скороченому вигляді алгоритми декодування ТК.

Для алгоритму *MAP* обчислення реберної (транзитної) метрики відбувається у формі

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

Вихідне *LLR MAP* декодера можна представити як суму трьох вимірювань: каналного, апіорного знання даних і зовнішнього декодера *LLR*, — у формі

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

де  $\ln(\cdot)$  — натуральний логарифм,  $\alpha_k(s) = \sum \gamma_k(s', s) \cdot \alpha_{k-1}(s')$  — метрика прямого шляху на діаграмі *RSC* кодера;  $\beta_{k-1}(s') = \sum \gamma_k(s', s) \cdot \beta_k(s)$  — метрика зворотного шляху на гратчастій діаграмі.

Модифікація *MAP* алгоритму під назвою *log-MAP* працює в логарифмічній області, має меншу обчислювальну складність і дає результати *BER*, які максимально наближені до результату *MAP*.

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

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

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

Функцію суми експоненціальних складових подано рекурсивно

$$f(A_1 \dots A_N) = \ln \sum_{i=1}^N e^{A_i} = f\left(A_1, f\left(A_2, \dots, f\left(A_{N-2}, f\left(A_{N-1}, A_N\right)\right)\right)\right). \quad (6)$$

Спрощення можна подати у формі логарифма з коректувальною функцією  $f_{cor}$

$$\begin{aligned} f(A_{N-1}, A_N) &= \ln(\exp(A_{N-1}) + \exp(A_N)) = \\ &= \max(A_{N-1}, A_N) + \ln(1 + \exp(-|A_{N-1} - A_N|)) = \max(A_{N-1}, A_N) + f_{cor}. \end{aligned} \quad (7)$$

Застосувавши формули (2)—(7), можна записати м'який вихід  $\log$ -MAP алгоритму в формі

$$\begin{aligned} LLR_{LM}(d_k) &= \left( \max_{d_k=+1}^{(s',s)} (\alpha_{k-1}^{LM}(s') + \beta_k^{LM}(s) + \gamma_k^{LM}(s',s)) + f_{cor.1} \right) - \\ &\quad - \left( \max_{d_k=-1}^{(s',s)} (\alpha_{k-1}^{LM}(s') + \beta_k^{LM}(s) + \gamma_k^{LM}(s',s)) + f_{cor.2} \right). \end{aligned} \quad (8)$$

Для економії часу, спрощення та швидкості обчислень, замість того, щоб декілька разів звертатися до відносно повільної та кошовної в апаратному виконанні функції  $\exp(\cdot)$ , для виразу  $f_{кор}$  використовують апроксимацію. Автори статті [14] пропонують використати ефективну модифікацію з кусково-лінійною апроксимацією  $PL$ - $\log$ -MAP.

Популярна модифікація алгоритму MAP під назвою  $max$ - $\log$ -MAP має набагато меншу обчислювальну складність, але втрачає властивість оптимальності, оскільки не використовується коректувальна складова. Ефективність для цього алгоритму на 10 % менше, в порівнянні з  $\log$ -MAP, але цей алгоритм є найменш складним з усіх існуючих методів [14]. Виникає необхідність доповнення алгоритму  $max$ - $\log$ -MAP за допомогою апроксимації компенсувальної функції з метою підвищення ефективності роботи декодера ТК.

Необхідно зазначити, що величини метрик можуть бути більше одиниці. Така особливість пов'язана з процесом рекурсивного обчислення та скороченням в процесі розрахунку транзитної метрики. Отже, для фрейму з великою кількістю бітів може виникнути проблема числового бар'єру внаслідок множення великих чисел. Тому метрики шляху назад та вперед нормують [10, 15].

### Алгоритм SOVA та його модифікація Bi-SOVA

Для роботи алгоритму декодування SOVA потрібно використати метрики ребер на решітці та блок операцій додати-порівняти-вибрати (ACS — Add-Compare-Select). Алгоритм декодування застосовує метрику шляху вперед, яка вибирається за допомогою операцій ACS до кінця решітки

$$M_k(s_k) = \max_{d_k=\pm 1}^{(s',s)} (M_{k-1}(s_{k-1}) + M_k(s',s)) = \max_{d_k=\pm 1}^{(s',s)} (M_{k-1}(s_{k-1}) + \gamma_k^{LM}(s',s)), \quad (9)$$

де  $M_{k-1}(s_{k-1})$  — поточна метрика у вершині трелліс-діаграми;  $M_k(s',s)$  — метрика ребра.

Потім застосовуємо апроксимацію J. Hagenauer [4—6, 11] на отриманому максимально правдоподібному шляху на решітці з ковзним вікном розміром  $\delta$  у вигляді

$$LLR_{SOVA} \approx d_k \cdot \min_{\substack{i=k \dots k+\delta \\ d_k \neq d_k^i}} \left( \max_{d_k=+1}^{(s',s)} (M_k^{d_k=+1}(s_k) - M_k^{d_k=-1}(s_k)) \right) = d_k \cdot \min_{\substack{i=k \dots k+\delta \\ d_k \neq d_k^i}} \Delta_i^{s_i}. \quad (10)$$

Модифікація такого алгоритму Bi-SOVA виконується вперед та назад за формулами (9), (10). Далі зберігаються значення  $M_k(s_k)$  вперед, назад та транзитна метрика  $M_k(s',s)$ , а також знаходяться величини м'якого виходу цього алгоритму декодування  $LLR_{Bi-SOVA}$  для біта даних

$$LLR_{Bi-SOVA}(d_k) = \begin{cases} LLR_{SOVA}^{\rightarrow}(d_k), & \text{if } LLR_{SOVA}^{\rightarrow}(d_k) > LLR_{SOVA}^{\leftarrow}(d_k); \\ LLR_{SOVA}^{\leftarrow}(d_k), & \text{if } LLR_{SOVA}^{\leftarrow}(d_k) > LLR_{SOVA}^{\rightarrow}(d_k). \end{cases} \quad (11)$$

де  $LLR_{SOVA}^{\rightarrow}(d_k)$ ,  $LLR_{SOVA}^{\leftarrow}(d_k)$  — апостеріорні м'які рішення, які знайдено відповідно за прямим та зворотним алгоритмом SOVA.

Також можна виконати нормалізацію метрик в станах, щоб зменшити числовий бар'єр.

### Результати досліджень складності реалізації алгоритмів декодування ТК на *DSP*

Будь-який *RSC* кодер описується решітчастою діаграмою, на якій кількість можливих станів  $T = 2^m$ , тоді загальна кількість переходів  $L = 2 \cdot S = 2^{m+1} = 2^K$ , де  $K$  — конструктивна довжина кодера (кодове обмеження).

Операцію ділення представимо як  $a \cdot (1/b)$ , експоненціювання набуде форму розкладу функції  $\exp(\cdot)$  в ряд на діапазоні  $(-\infty; +\infty)$ , а логарифмування розглянемо як ряд для  $\ln(\cdot)$  на діапазоні  $(0; +\infty)$  [18]. Врахуємо, що  $d^{hard} = \text{sign}(LLR_{apost})$ , а величина  $n(\delta)$  — це кількість операцій мінімуму  $n$  на ковзному вікні розміром  $\delta$  (нехай  $n(\delta) = 5 \cdot K = 5 \cdot (m + 1) - 1 = 5 \cdot m + 4$  [4—6]). Отже, всі операції будуть відповідати ЕМО для *DSP*. Таким чином, враховуючи написане вище, відобразимо в таблицях 1—5 аналітичні вирази для визначення кількості ЕМО для декодування біта інформації на *DSP* за використання різних алгоритмів декодування ТК.

Таблиця 1

Кількість ЕМО для декодування біта інформації на *DSP* за алгоритмом *MAP*

ЕМО	Параметри алгоритму декодування <i>MAP</i>							
	$\gamma(s', s)$	$\alpha_k(s)$	$\beta_{k-1}(s')$	$\tilde{\alpha}_k(s)$	$\tilde{\beta}_{k-1}(s')$	$LLR_{MAP}$	$LLR_{MAP}^{ext}$	$d_k^{hard}$
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

Таблиця 2

Кількість ЕМО для декодування біта інформації на *DSP* за алгоритмом *PL-log-MAP*

ЕМО	Параметри алгоритму декодування <i>PL-log-MAP</i>							
	$\gamma(s', s)$	$\alpha_k(s)$	$\beta_{k-1}(s')$	$\tilde{\alpha}_k(s)$	$\tilde{\beta}_{k-1}(s')$	$LLR_{PL-log-MAP}$	$LLR_{PL-log-MAP}^{ext}$	$d_k^{hard}$
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

Таблиця 3

Кількість ЕМО для декодування біта інформації на *DSP* за алгоритмом *max-log-MAP*

ЕМО	Параметри алгоритму декодування <i>max-log-MAP</i>							
	$\gamma(s', s)$	$\alpha_k(s)$	$\beta_{k-1}(s')$	$\tilde{\alpha}_k(s)$	$\tilde{\beta}_{k-1}(s')$	$LLR_{MLM}$	$LLR_{MLM}^{ext}$	$d_k^{hard}$
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

Кількість ЕМО для декодування біта інформації на *DSP* за алгоритмом *SOVA*

ЕМО	Параметри алгоритму декодування <i>SOVA</i>						
	$\gamma(s', s)$	$M_k(s_k)$	$\tilde{M}_k(s_k)$	$\Delta_i^{s_i}$	$LLR_{SOVA}$	$LLR_{SOVA}^{ext}$	$d_k^{hard}$
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

Кількість ЕМО для декодування біта інформації на *DSP* за алгоритмом *Bi-SOVA*

ЕМО	Параметри алгоритму декодування <i>Bi-SOVA</i>						
	$\gamma(s', s)$	$M_k(s_k)$	$\tilde{M}_k(s_k)$	$\Delta_i^{s_i}$	$LLR_{Bi-SOVA}$	$LLR_{Bi-SOVA}^{ext}$	$d_k^{hard}$
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

Додавши всі операції для декодування біта інформації, отримуємо загальну кількість ЕМО для кожного з розглянутих алгоритмів декодування. Отже, складність декодування біта інформації за показником ЕМО для відповідних алгоритмів декодування представляється функцією  $f(m, q)$ , максимум якої в  $(+\infty; +\infty)$ , а мінімум в точці [1; 2]. Отримали оцінку кількості ЕМО, які необхідно виконати *DSP* для декодування біта інформації при використанні компонентних *RSC* кодерів з різною кількістю перевірочних бітів на виході

$$f_{MAP}(m, q) = 2^{m+2} \cdot q + 153 \cdot 2^{m+1} + 361; \quad (13)$$

$$f_{PL-\log-MAP}(m, q) = 2^{m+2} \cdot q + 33 \cdot 2^{m+1} - 19; \quad (14)$$

$$f_{MLM}(m, q) = 2^{m+2} \cdot q + 11 \cdot 2^{m+1} + 3; \quad (15)$$

$$f_{SOVA}(m, q) = 2^{m+2} \cdot q + 11 \cdot 2^m + 5 \cdot m + 11; \quad (16)$$

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

Залежність  $f$  складності розглянутих алгоритмів декодування від кількості комірок пам'яті  $m$  *RSC* кодера для різної кількості бітів  $q$  на виході кодера показана на рис. 1. Аналіз табл. 1—5, формул (13)—(17) та рис. 1 свідчить про те, що складність реалізації алгоритмів декодування ТК зі збільшенням  $m$  *RSC* кодера зростає у вигляді степеневі функції, а з ростом величини  $q$  значення функції збільшується на показник  $2^{m+2}$ . Крім того, можна помітити, що за однакових вихідних даних найскладнішим є *MAP* алгоритм, далі *PL-log-MAP*, *max-log-MAP*, *Bi-SOVA*, а найменш складний — *SOVA*. Для порівняльного аналізу алгоритмів декодування ТК скористаємося параметрами відносної складності  $O_i$  (рис. 2), які показують у скільки разів один алгоритм складніший за інший, що подамо у формі

$$O_i \cdot n_{iEMO} = N_{iEMO}, \quad (18)$$

де  $N_{iEMO}$ ,  $n_{iEMO}$  — відповідно, більша та менша кількості ЕМО для алгоритмів, які порівнюються.

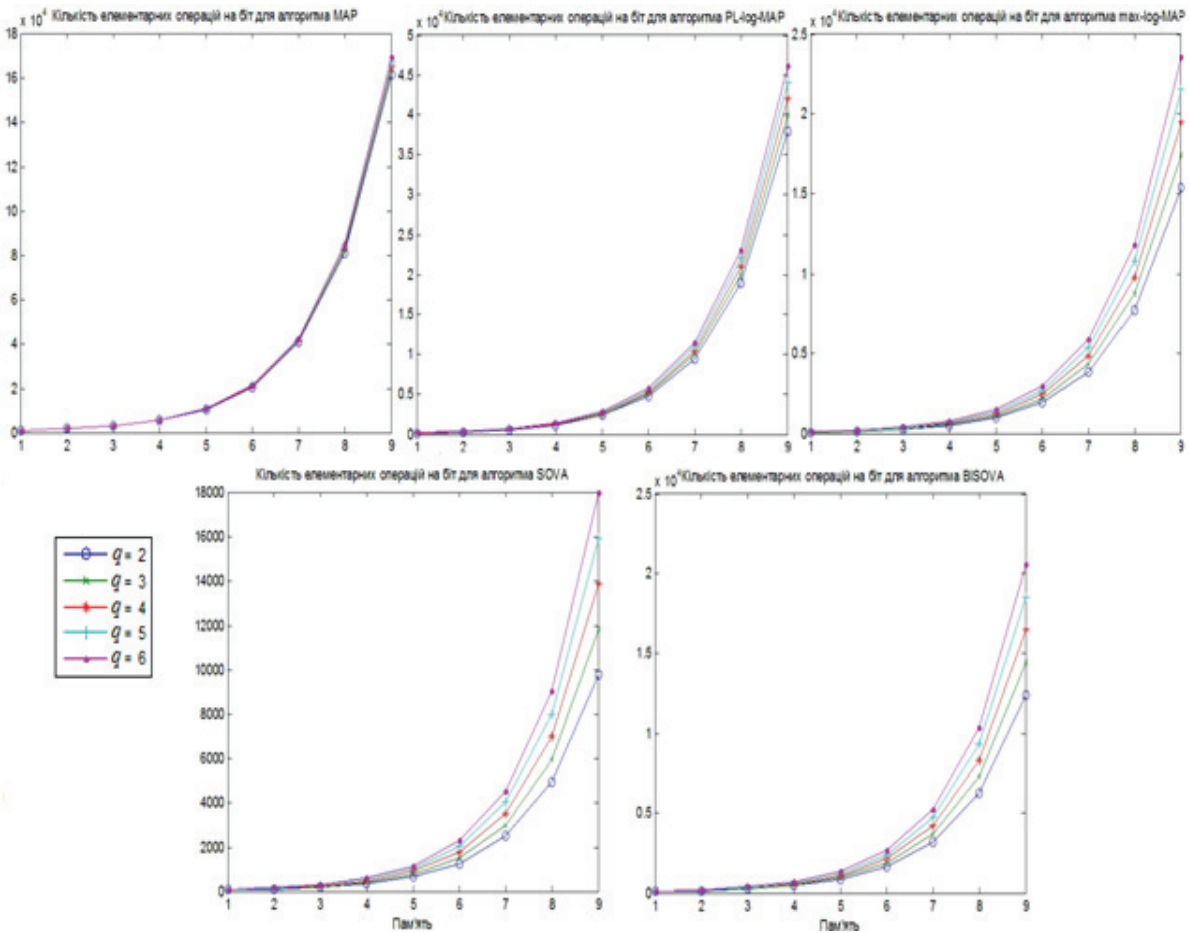


Рис. 1. Загальна кількість ЕМО як функція  $f(m, q)$  для розглянутих алгоритмів під час декодування біта інформації на DSP

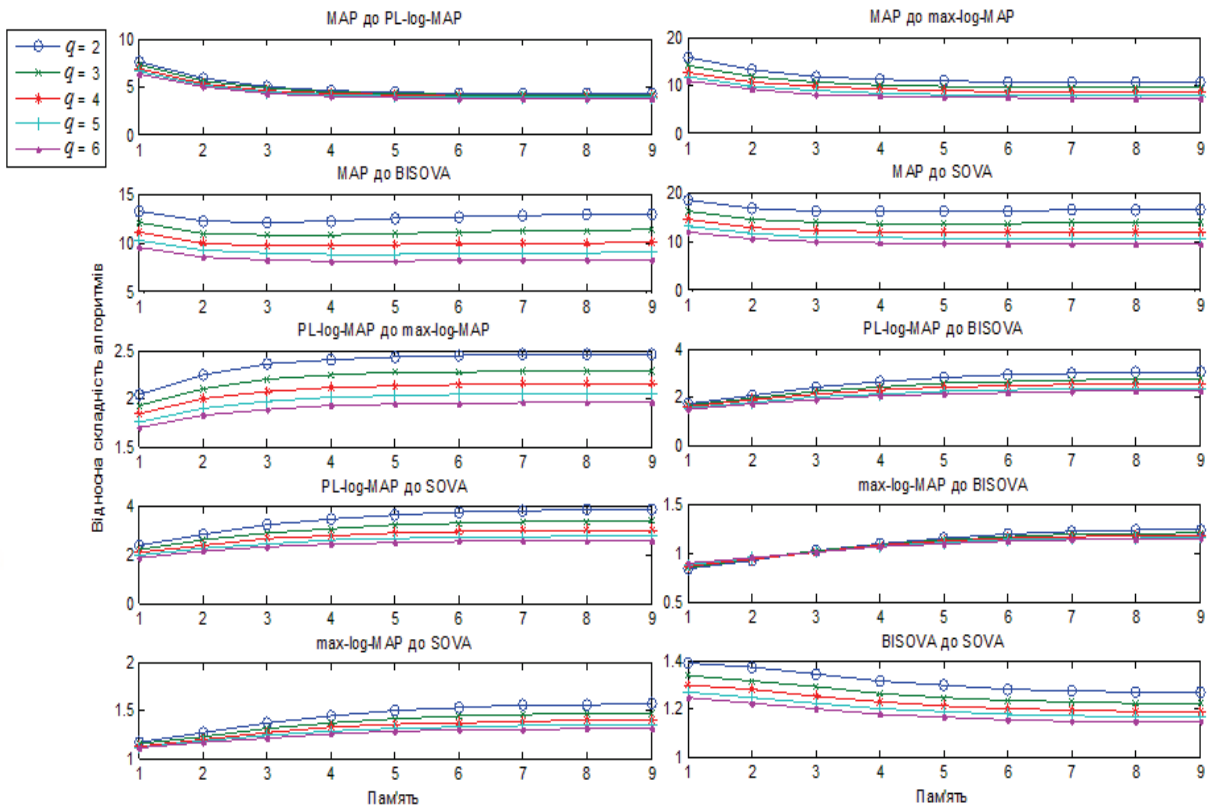


Рис. 2. Відносна складність різних алгоритмів декодування ТК

Аналіз рис. 2 та формули (18) свідчить про те, що кількість ЕМО зі збільшенням  $m$  в алгоритмі *MAP* росте повільніше, ніж в інших алгоритмах ( $O_i$  спадає), крім *SOVA* та *Bi-SOVA* ( $O_i$  зростає). Алгоритм *MAP* складніший за *PL-log-MAP* в 3,68...7,67 разів, за *max-log-MAP* в 7,19...15,7 разів, за *SOVA* в 9,42...18,31 разів та за *Bi-SOVA* в 8,22...13,19 разів. Подібно до цього найрізкіше росте кількість ЕМО в алгоритмі *PL-log-MAP*, тому зі збільшенням  $m$  відносна складність зростає, прямуючи до  $+\infty$ . Можна сказати, що алгоритм *PL-log-MAP* складніший в 1,69...2,46 разів за *max-log-MAP*, в 1,5...3,06 разів — за *Bi-SOVA*, 1,87...3,87 разів — за *SOVA*. Кількість ЕМО *max-log-MAP* росте швидше, ніж *Bi-SOVA* та *SOVA*. Кількість ЕМО в алгоритмі *max-log-MAP* росте швидше в 1,1...1,57 разів, ніж у *SOVA*, а також в 0,84...1,23 рази швидше за *Bi-SOVA*. Алгоритм *Bi-SOVA* в 1,24...1,39 рази складніший за *SOVA*, а кількість ЕМО в ньому росте повільніше за *SOVA*.

### Висновки

З врахуванням складності задачі декодування та реалізації турбо-декодера, визначені певні особливості, які ведуть до її розв'язання. Описані алгоритми, які використовуються для декодування ТК. Встановлено та розглянуто функціональні залежності  $f(m, q)$ , які визначають кількість ЕМО (з урахуванням кількості комірок пам'яті і загальної кількості символів з виходу *RSC* кодера) для різних алгоритмів декодування ТК.

На використанні ітеративного декодера, який складається з компонентних *SISO* декодерів, базуються всі описані алгоритми декодування ТК, такі як оптимальні для декодування найімовірнішого біта інформації *MAP* та *log-MAP*, субоптимальні *PL-log-MAP* та *max-log-MAP*, а також оптимальний для декодування найімовірнішої послідовності *SOVA* та його модифікація *Bi-SOVA*. Алгоритм *MAP* є надзвичайно складним в обчислювальному плані, *log-MAP* спрощує алгоритм *MAP* і пропонує оптимальну ефективність з допустимим рівнем складності. Ігнорування компенсальної функції оригінального *log-MAP* алгоритму дозволяє застосувати просту грубу модифікацію для зниження обчислювальної складності — *max-log-MAP*. Цей алгоритм субоптимальний з втратою ефективності, саме тому коректувальну функцію апроксимують зручними для роботи виразами, що дозволяє застосувати різні модифікації, наприклад *PL-log-MAP* алгоритм, який має ефективність близьку до *log-MAP*, але є простішим в обчислювальному плані.

Аналітичні вирази для оцінювань складності процесу роботи декодера, отримані в роботі, можуть бути використані для подальшого аналізу складності ТК у разі його апаратно-програмної реалізації. Тому для порівняння складності реалізації алгоритмів декодування ТК ЕМО, необхідні для декодування біта інформації, повинні бути представлені як елементарні для *DSP*. Отримані вирази доцільно використовувати для аналізу складності реалізації ТК на *DSP* з метою вибору елементної бази.

### СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Berrou C. Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes / C. Berrou, A. Glavieux, P. Thitimajshima // Proceedings of the ICC'93. — Switzerland, Geneva, 1993. — P. 1064—1070.
2. Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate / L. Bahl, J. Cocke, F. Jelinek, J. Raviv // IEEE Transactions on Information Theory. — 1974. — V. 20. — P. 284—287.
3. Иванов Ю. Ю. О некоторых аспектах итеративной стратегии декодирования турбо-кодов: ретроспектива и «турбо»-принцип. / Ю. Ю. Иванов, А. Я. Кулик. // Информационные технологии и компьютерная инженерия (ИТКИ) : IV международной научно-практической конференции : матер. — Винница : ВНТУ, 28—30 мая, 2014 года. — С. 157—160.
4. Hagenauer J. A Viterbi Algorithm with Soft-Decision Outputs and its Applications / J. Hagenauer, P. Hoeher // Proceedings of IEEE Global Telecommunications Conference. — USA, Texas: 1989. — P. 1680—1686.
5. Woodard J. Comparative Study of Turbo Decoding Techniques: An Overview / J. Woodard, L. Hanzo // IEEE Transactions on Vehicular Technology. — 2000. — V. 49, № 6. — P. 2208—2233.
6. Hanzo L. Turbo Coding, Turbo Equalisation and Space-Time Coding for Transmission over Wireless Channels / L. Hanzo, T. H. Liew, B. L. Yeap. — Southampton: Department of Electronics and Computer Science of UK, 2002. — 746 p.
7. Morelos-Zaragoza R. The Art of Error Correction Coding: 2nd Edition / Morelos-Zaragoza R. — Chippenham: John Wiley & Sons, Ltd, 2006. — P. 143—168.
8. Топалов В. В. Модификация двунаправленного итеративного алгоритма Витерби с вероятностным решением для декодирования турбо-кодов / В. В. Топалов // Наукові праці ОНАЗ ім. О. С. Попова. — 2007. — № 2. — С. 74—75.
9. Robertson P. A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain / P. Robertson, E. Villebrun, P. Hoeher // Gateway to Globalization : 1995 IEEE International Conference on Communications ICC'95. — USA, Seattle: IEEE, June 18, 1995. — V. 2. — P. 1009—1013.
10. Abrantes S. From BCJR to Turbo Decoding: MAP Algorithms Made Easier / S. Abrantes // Information and Telecommunication Technology Center of the University of Kansas, USA. — Lawrence : ITTC, 2004. — 30 p.
11. Soleymani M. R. Turbo Coding for Satellite and Wireless Communications / M. R. Soleymani, Y. Gao, U. Vilaipornswai. — New York: Kluwer Academic, 2002. — 231 p.

12. Channel Coding: Theory, Algorithms, and Applications / [D. Declercq, M. Fossorier, Ezio Biglieri and others]. — Academic Press Library in Mobile and Wireless Communications, Elsevier, 2014. — 690 p.

13. Модификации для снижения вычислительной сложности алгоритма декодирования BCJR MAP в турбо-кодowych конструкциях / [О. В. Стукач, А. Н. Романюк, А. Я. Кулик, Ю. Ю. Иванов] // Наукові праці ДонНТУ. — 2015. — № 1 (20). — С. 107—112. — (Інформатика, кібернетика та обчислювальна техніка).

14. A Novel Suboptimal Piecewise-Linear-log-MAP Algorithm for Turbo Decoding / Yu. Yu. Ivanov, A. N. Romanyuk, A. Ia. Kulyk, O. V. Stukach // Proceedings on XI IEEE International Siberian Conference on Control and Communications (SIBCON-2015). — Omsk : Omsk State Technical University, Russia, 21—23 May, 2015. — P. 1—8.

15. Ливенцев С. П. Анализ сложности MAP, MAX LOG MAP и LOG MAP алгоритмов декодирования турбокодов при декодировании бита информации / С. П. Ливенцев, С. В. Зайцев, Б. В. Горлинский // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. — 2006. — № 1 (12). — С. 125—135.

16. Зайцев С. В. Оценка сложности реализации алгоритмов декодирования турбокодов при декодировании бита информации на цифровых сигнальных процессорах / С. В. Зайцев, С. П. Ливенцев, Б. В. Горлинский // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. — 2006. — № 2 (13). — С. 183—188.

17. Кулик А. Я. Порівняльний аналіз складності реалізації методів декодування турбо-кодів / А. Я. Кулик, С. Г. Кривошубченко, Ю. Ю. Иванов // Інформаційні технології та комп'ютерна інженерія. — 2013. — № 1 (26). — С. 26—31.

18. Бронштейн И. Н. Справочник по математике для инженеров и учащихся ВТУЗов / И. Н. Бронштейн, К. А. Семенов. — 9-е изд. — М. : Государственное изд-во физико-математической литературы, 1962. — 608 с.

Рекомендована Центром інформаційних технологій і захисту інформації ВНТУ

Стаття надійшла до редакції 4.04.2016

**Іванов Юрій Юрійович** — аспірант кафедри автоматики та інформаційно-виміральної техніки, e-mail: Yura881990@yandex.ru.

Вінницький національний технічний університет, Вінниця

**Yu. Yu. Ivanov<sup>1</sup>**

## Hardware and Software Features of Turbo-Codes: Analysis of Implementation Complexity on Digital Signal Processor

<sup>1</sup>Vinnitsia National Technical University

*The article provides a brief overview of the algorithms used for encoding turbo-codes. There have been produced the complexity assessment of the implementation for the decoding information bit during the turbo-code hardware and software implementation. There have been established and analyzed functional dependence on the memory and the total number of bits after coders for the different decoding algorithms.*

**Keywords:** error-correcting coding, turbo-code, decoding algorithms, MAP, log-MAP, PL-log-MAP, max-log-MAP, Bi-SOVA, SOVA, number of elementary mathematical operations, digital signal processors.

**Ivanov Yuriy Yu.** — Post-Graduate Student of the Chair of Automation and Informational-Measuring Technique, e-mail: Yura881990@yandex.ru

**Ю. Ю. Иванов<sup>1</sup>**

## Особенности аппаратно-программной реализации турбо-кодов: анализ сложности реализации на цифровом сигнальном процессоре

<sup>1</sup>Вінницький національний технічний університет

*Проведен обзор алгоритмов, используемых для кодирования турбо-кодов. Дана оценка сложности реализации декодирования бита информации в ходе аппаратно-программной реализации турбо-кода. Установлена и проанализирована функциональная зависимость разных алгоритмов декодирования турбо-кодов от памяти и общего количества битов с декодеров.*

**Ключевые слова:** помехоустойчивое кодирование, турбо-код, алгоритмы декодирования, MAP, log-MAP, PL-log-MAP, max-log-MAP, Bi-SOVA, SOVA, количество элементарных математических операций, цифровые сигнальные процессоры.

**Іванов Юрій Юрійович** — аспірант кафедри автоматики та інформаційно-виміральної техніки, e-mail: Yura881990@yandex.ru