

УДК 621.391

Liliia Zaitseva, Postgraduate student

ORCID ID: <https://orcid.org/0000-0002-0668-711X> **e-mail:** lili5990n@ukr.net

Institute of Telecommunications and Global Information Space of NAS of Ukraine, Kyiv, Ukraine

METHOD OF PARAMETRIC ADAPTATION OF PARALLEL AND SEQUENTIAL TURBO CODES USING NEURAL NETWORKS

Abstract. *The work is devoted to the study of increasing the efficiency of functioning of modern wireless technologies 5G and 6G. The article presents a method of parametric adaptation of parallel and sequential turbo codes using neural networks of the multilevel perceptron type and the decoding uncertainty indicator.*

The use of neural networks of the multilevel perceptron type for adjusting the external logarithmic ratios of the likelihood functions of probabilistic algorithms for decoding parallel and sequential turbo codes is considered.

Turbo codes are decoded using the maximum a posteriori probabilities (MAP) decoding algorithm, which calculates the posterior probability of each decoded symbol, minimizing the probability of an information symbol (bit) error.

The aim of the work is to develop a method for parametric adaptation of parallel and sequential turbo codes using neural networks of the multilevel perceptron type and a decoding uncertainty indicator.

The use of the decoding uncertainty indicator for parallel and sequential turbo codes at the training stage when determining the weight coefficients of the weight matrix and when functioning of neural networks is proposed.

Keywords: 5G, 6G, wireless technologies, turbo codes, neural networks, multi-layer perceptron, decoding uncertainty.

Л.І. Зайцева

Інститут телекомунікацій і глобального інформаційного простору НАН України, м. Київ, Україна

МЕТОД ПАРАМЕТРИЧНОЇ АДАПТАЦІЇ ПАРАЛЕЛЬНИХ ТА ПОСЛІДОВНИХ ТУРБО КОДІВ З ВИКОРИСТАННЯМ НЕЙРОННИХ МЕРЕЖ

Анотація. *Роботу присвячено дослідженню підвищення ефективності функціонування сучасних безпроводних технологій 5G та 6G. В статті представлено метод параметричної адаптації паралельних та послідовних турбо кодів з використанням нейронних мереж типу багаторівневий перцептор та показника невизначеності декодування.*

Розглянуто використання нейронних мереж типу багаторівневий перцептор для коригування зовнішніх логарифмічних відношень функцій правдоподібності ймовірнісних алгоритмів декодування паралельних та послідовних турбо кодів. Декодування турбо кодів відбувається за допомогою алгоритму декодування по максимуму апостеріорної ймовірності MAP (maximum a posteriori probabilities), який здійснює розрахунок апостеріорної ймовірності кожного декодованого символу, мінімізуючи ймовірність помилки інформаційного символу (біта).

Метою роботи є розробка методу параметричної адаптації паралельних та послідовних турбо кодів з використанням нейронних мереж типу багаторівневий перцептрон та показника невизначеності декодування. Запропоновано використання показника невизначеності декодування паралельних та послідовних турбо кодів на етапі навчання при визначенні вагових коефіцієнтів матриці ваг та при функціонуванні нейронних мереж.

Ключові слова: 5G, 6G, безпроводні технології, турбо коди, нейронні мережі, багаторівневий перцептрон, невизначеність декодування.

<https://doi.org/10.32347/2411-4049.2026.1.214-224>

Вступ

Безпроводовий зв'язок має вирішальне значення в багатьох секторах, таких як телекомунікації, аерокосмічна галузь, оборона, охорона здоров'я, Інтернет речей та побутова електроніка. Ці системи продовжують розвиватися від покоління до покоління. Наразі використовуються безпроводові системи 5G, і досліджуються перспективні – мережі 6G або Beyond 5G (B5G) [1-3]. Ці майбутні системи спрямовані на забезпечення наднадійного, низькозатримкового, високопродуктивного та штучно-інтелектуального з'єднання. Досягнення цих цілей поставило нові виклики, які виходять за межі можливостей традиційних методів кодування каналів, таких як коди з низькою щільністю перевірки на парність (LDPC) та турбо коди.

Цілями безпроводових комунікаційних систем наступного покоління, особливо 6G, є покращене з'єднання, зменшення затримки та вища швидкість передачі даних. Для систем 6G досліджуються схеми кодування квадратурної амплітудної модуляції (QAM), використання квазіциклічних кодів з малою перевіркою на парність (QC-LDPC), модуляції QAM з турбо кодами, модуляції QAM з LDPC-кодами та модуляції QAM з полярними кодами [4-5].

Метою роботи є розробка методу параметричної адаптації паралельних та послідовних турбо кодів з використанням нейронних мереж типу багаторівневий перцептрон та показника невизначеності декодування.

Виклад основного матеріалу дослідження

Для покращення характеристик ЛВФП турбо кодів при паралельному та послідовному з'єднанні РСЗК, будемо використовувати MLP-нейронну мережу (MLP – Multilayer Perceptron, багатосаровий перцептрон).

Запропоновано алгоритм корекції ЛВФП через MLP для турбо кодів.

1. Ініціалізація.

Для пакету t :

1) Отримуємо з каналу прийняті символи u_t .

2) Ініціалізуємо апріорні ЛВФП (якщо перший декодер турбо коду):

$$L_a^{(0)}(u_k) = 0 \quad \forall k, \quad (1)$$

де u_k – k -й інформаційний біт.

2. Обчислення зовнішнього ЛВФП декодера турбо коду.

Декодування турбо кодів відбувається за допомогою алгоритму декодування по максимуму апостеріорної ймовірності MAP (maximum a posteriori probabilities), який здійснює розрахунок апостеріорної ймовірності кожного декодованого символу, мінімізуючи ймовірність помилки

інформаційного символу (біта). Декодування турбо кодів відбувається за такою ж решітчастою діаграмою, за якою працює кожен RSCC турбо коду.

Наприклад, алгоритм декодування MAP двох біт турбо коду для виду RSCC (1, 5/7) показаний на рис. 1.

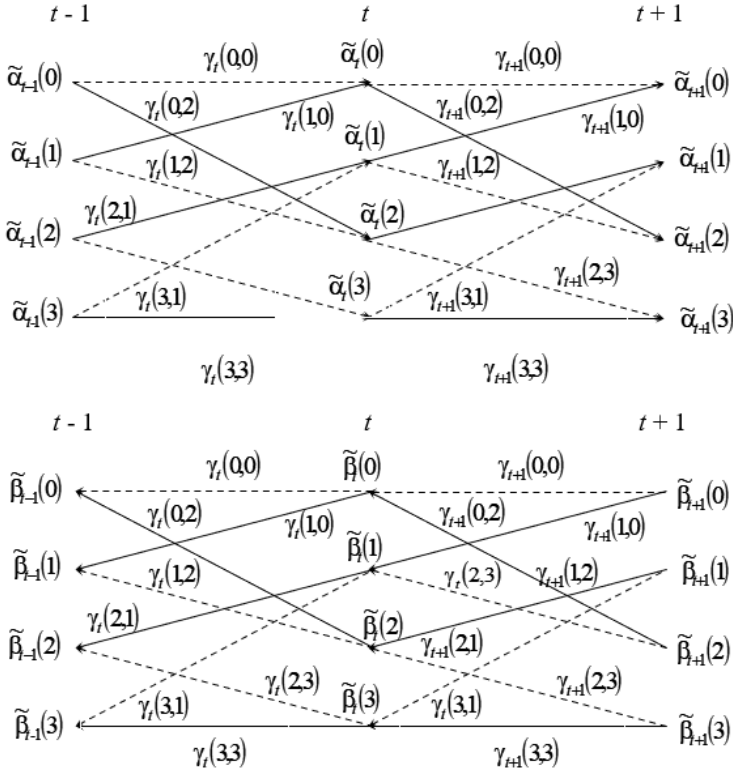


Рис. 1. Алгоритм декодування MAP двох біт турбо коду для виду RSCC (1, 5/7)

Декодування відбувається за двома напрямками: у першому напрямку (від початку блоку до кінця) для кожного стану турбо коду обчислюються прямі та перехідні рекурсії, у другому напрямку (з кінця блоку до початку) відбувається розрахунок зворотних рекурсій, використовуючи перехідні рекурсії, отримані при першому напрямку обчислень.

Логарифмічне відношення функцій правдоподібності про переданий біт $L(u_i)$ залежить від каналної інформації $L_c(y_i)$, апіорної інформації про переданий біт $L_a(x_i)$ та апостеріорного LLR, виробленого безпосередньо самим декодером $L_e(x_i)$. Тому при декодуванні біта u_i для обчислень першим декодером на ітерації декодування $j, j \in \overline{1, I}$, де I – загальна кількість ітерацій декодування, вираз LLR може бути записано таким чином:

$$L^{1,j}(x_i) = \log \frac{\sum_{\substack{(s',s) \\ u_i=1}} \tilde{\alpha}_{t-1}^{(1)}(s') \cdot \tilde{\beta}_t^{(1)}(s) \cdot \gamma_t^{(1)}(s',s)}{\sum_{\substack{(s',s) \\ u_i=0}} \tilde{\alpha}_{t-1}^{(1)}(s') \cdot \tilde{\beta}_t^{(1)}(s) \cdot \gamma_t^{(1)}(s',s)} = L_c^{1,j}(y_i) + L_a^{1,j}(x_i) + L_e^{1,j}(x_i), \quad (2)$$

де $L_c^{1,j}(y_t)$ – канална інформація, $L_e^{1,i}(x_t)$ – апостеріорне LLR біту даних x_t , $\tilde{\alpha}_{t-1}^{(1)}(s')$ – нормалізоване значення прямої рекурсії $\alpha_{t-1}^{(1)}(s')$, $\tilde{\beta}_t^{(1)}(s)$ – нормалізоване значення зворотної рекурсії $\beta_t^{(1)}(s)$.

Відповідно, для другого декодера отримаємо:

$$L^{2,j}(x_t) = \log \frac{\sum_{\substack{(s',s) \\ u_t=1}} \tilde{\alpha}_{t-1}^{(2)}(s') \cdot \tilde{\beta}_t^{(2)}(s) \cdot \gamma_t^{(2)}(s',s)}{\sum_{\substack{(s',s) \\ u_t=0}} \tilde{\alpha}_{t-1}^{(2)}(s') \cdot \tilde{\beta}_t^{(2)}(s) \cdot \gamma_t^{(2)}(s',s)} = L_c^{2,j}(y_t) + L_a^{2,j}(x_t) + L_e^{2,j}(x_t). \quad (3)$$

Далі розраховується апостеріорне LLR біта даних, вироблене самим декодером, $-L_e^{1,j}(x_t)$:

$$L_e^{1,j}(x_t) = L^{1,j}(x_t) - L_c^{1,j}(y_t) - L_a^{1,j}(x_t). \quad (4)$$

Після перемержувача Π апостеріорне LLR $L_e^{1,i}(x_t)$ перетворюється на апіорне LLR $L_a^{2,j}(x_t)$: $L_a^{2,j}(x_t) = f_1(L_e^{1,i}(x_t))$, де $f_1(\cdot)$ – функція, що здійснює операції перемержування, і подається на декодер 2. Декодер 2 виконує аналогічні обчислення для отримання величини $L_e^{2,i}(x_t)$:

$$L_e^{2,j}(x_t) = L^{2,j}(x_t) - L_c^{2,j}(y_t) - L_a^{2,j}(x_t). \quad (5)$$

Виконавши операцію деперемержування D : $L_a^{1,j+1}(x_t) = f_2(L_e^{2,i}(x_t))$, де $f_2(\cdot)$ – функція, що здійснює операції деперемержування, величина використовується як апіорна для декодера 1 ітерації $j + 1$. Далі здійснюються обчислення, аналогічні (2), (3). Після виконання всіх ітерацій декодування виносяться «жорсткі» рішення про переданий біт: $\tilde{u}_t = \text{sign}[L(u_t)]$.

З урахуванням HARQ основні математичні співвідношення запишуться наступним чином:

Розрахунок LLR біта даних x_t , $t \in \overline{1, N}$, i -м декодером, $i \in \overline{1, 2}$, j -ї ітерації декодування, $j \in \overline{1, I}$, для всіх біт блоку довжиною N , декодера 1 і 2, ітерацій декодування $j \in \overline{1, I}$, де I – загальна кількість ітерацій декодування, h – параметр автоматичних запитів на повторну передачу.

$$L^{i,j,h}(x_t) = \log \frac{\sum_{\substack{(s',s) \\ u_t=1}} \tilde{\alpha}_{t-1}^{(i)}(s') \cdot \tilde{\beta}_t^{(i)}(s) \cdot \gamma_t^{(i)}(s',s)}{\sum_{\substack{(s',s) \\ u_t=0}} \tilde{\alpha}_{t-1}^{(i)}(s') \cdot \tilde{\beta}_t^{(i)}(s) \cdot \gamma_t^{(i)}(s',s)} =$$

$$\log \frac{\sum_{\substack{(s',s) \\ u_t=1}} \tilde{\alpha}_{t-1}^{(i)}(s') \cdot \tilde{\beta}_t^{(i)}(s) \cdot \exp \left[\frac{1}{2} \cdot \left(x_t^{C,h} \cdot \left(L_a^{i,j,h}(x_t^{C,h}) + L_c \cdot y_t^{C,h} \right) + L_c \cdot y_t^{\Pi i,h} x_t^{\Pi i,h} \right) \right]}{\sum_{\substack{(s',s) \\ u_t=0}} \tilde{\alpha}_{t-1}^{(i)}(s') \cdot \tilde{\beta}_t^{(i)}(s) \cdot \exp \left[\frac{1}{2} \cdot \left(x_t^{C,h} \cdot \left(L_a^{i,j,h}(x_t^{C,h}) + L_c \cdot y_t^{C,h} \right) + L_c \cdot y_t^{\Pi i,h} x_t^{\Pi i,h} \right) \right]} =$$

$$= L_c^{i,j,h}(y_t) + L_a^{i,j,h}(x_t) + L_e^{i,j,h}(x_t).$$

Розрахунок апостеріорного LLR біта даних x_t , $t \in \overline{1, N}$ i -м декодером, $i \in \overline{1, 2}$, j -й ітерації декодування, $j \in \overline{1, I}$, для всіх бітів блока довжиною N , декодера 1 і 2, ітерацій декодування $j \in \overline{1, I}$:

$$L_e^{i,j,h}(x_t) = L^{i,j,h}(x_t) - L_c^{i,j,h}(y_t) - L_a^{i,j,h}(x_t).$$

Якщо існують помилки в прийнятому блоці довжиною N , формується сигнал HARQ, який передається на декодер для модифікації алгоритму декодування і надходить по каналу зворотного зв'язку для повторної передачі блоку даних. Параметр автоматичних запитів зворотної передачі $h = h + 1$.

3. Формування ознак для MLP.

Для кожного біту u_k створюється feature vector:

$$f_k = [L_e^{(i)}(u_k), |L_e^{(i-1)}(u_k)|, \Delta L_e(u_k), N^{\text{о}} \text{ ітерації, CRC, HARQ}]. \tag{6}$$

– Розраховується $\Delta L_e(u_k) = L_e^{(i)}(u_k) - L_e^{(i-1)}(u_k)$ – зміна ЛВФП на попередній ітерації.

4. Корекція LLR через MLP.

MLP навчається як коригувальна функція, яка перетворює зовнішнє ЛВФП:

$$\tilde{L}_e(u_k) = f_{\theta}(f_k), \tag{7}$$

де:

– f_{θ} – MLP з параметрами θ .

– Вихід: скоригована зовнішня ЛВФП для біта u_k .

MLP має наступну архітектуру: Input \rightarrow Dense64 \rightarrow ReLU \rightarrow Dense32 \rightarrow ReLU \rightarrow Dense1 \rightarrow Identity.

Розглянемо цю архітектуру детальніше.

4.1. Вхідний шар (input).

Нехай вхідний вектор:

$$x = [x_1, x_2, \dots, x_N] \in \mathbb{R}^N,$$

де елементами є LLR значення, прийняті з каналу з урахуванням «канальної» надійності.

4.2. Перший повнозв'язний шар (Dense64) (рис. 2).

Вхідний вектор

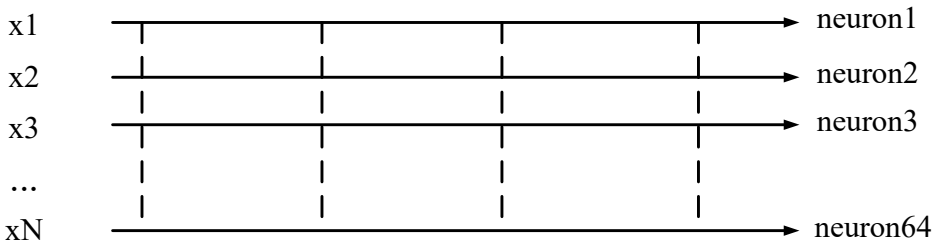


Рис. 2. Перший повнозв'язний шар (Dense64)

Тут відбувається лінійне перетворення:

$$z_1 = W_1 x + b_1,$$

де

$$W_1 \in \mathbb{R}^{64 \times N},$$

$$b_1 \in \mathbb{R}^{64},$$

$$z_1 \in \mathbb{R}^{64}.$$

У цьому виразі W_1 – матриця параметрів, які навчаються під час тренування мережі.

Розмірність:

$$W_1 \in \mathbb{R}^{64 \times N},$$

де N – кількість входів, 64 – кількість нейронів у шарі Dense 64.

Кожен нейрон має свій набір ваг.

Наприклад, для нейрона i :

$$w_i = [w_{i1}, w_{i2}, \dots, w_{iN}].$$

Ці ваги визначають важливість кожного входу.

Далі, b_1 – вектор зміщення Bias, це додатковий параметр для кожного нейрона.

Розмірність:

$$b_1 \in \mathbb{R}^{64},$$

Тобто

$$b_1 = [b_1^{(1)}, b_1^{(2)}, \dots, b_1^{(64)}].$$

Результат лінійного перетворення z_1 – це вихід лінійної частини шару (до активації).

Розмірність:

$$z_1 \in \mathbb{R}^{64},$$

Обчислення для нейрона i :

$$z_{1,i} = w_{i1}x_1 + w_{i2}x_2 + \dots + w_{iN}x_N + b_i.$$

Тобто, це скалярний добуток входу і ваг плюс bias.

Повнозв'язний (Dense) шар означає, що кожен вхід з'єднаний з кожним нейроном шару. У шарі Dense64 є 64 нейрони.

Шар обчислює 64 таких нейрони одночасно:

$$z_1 = \begin{bmatrix} z_1^{(1)} \\ z_1^{(2)} \\ \dots \\ z_1^{(64)} \end{bmatrix}$$

або

$$z_1 = W_1 x + b_1,$$

де

$$W_1 = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1N} \\ w_{21} & w_{22} & \dots & w_{2N} \\ \dots & \dots & \dots & \dots \\ w_{64,1} & w_{64,2} & \dots & w_{64,N} \end{bmatrix},$$

4.3. Активация ReLU (Rectified Linear Unit – випрямлений лінійний блок).
Функція

$$\text{ReLU}(z) = \max(0, z).$$

Тобто

$$h_1 = \text{ReLU}(z_1)$$

або

$$h_1 = \begin{cases} z_{1i}, & z_{1i} > 0 \\ 0, & z_{1i} < 0 \end{cases},$$

$$h_1 \in \mathbb{R}^{64}.$$

4.4. Другий Dense шар (Dense32) (рис. 3).

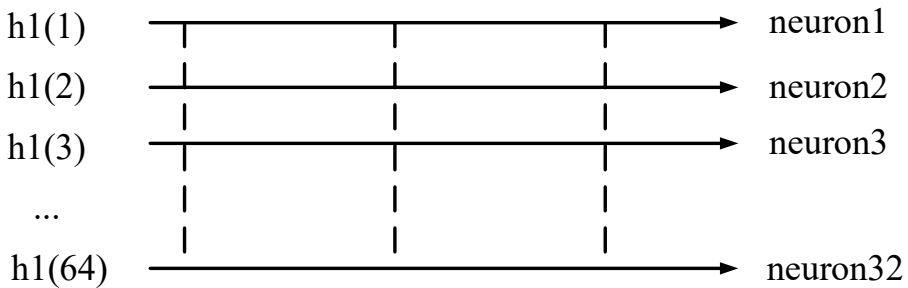


Рис. 3. Другий Dense шар (Dense32)

Використовується наступне лінійне перетворення:
Тут відбувається лінійне перетворення:

$$z_2 = W_2 h_1 + b_2,$$

де

$$W_2 \in \mathbb{R}^{32 \times 64},$$

$$b_2 \in \mathbb{R}^{32},$$

$$z_2 \in \mathbb{R}^{32}.$$

Після шару Dense64 та ReLU отримуємо вектор з 64 значень:

$$h_1 = [h_1^{(1)}, h_1^{(2)}, \dots, h_1^{(64)}].$$

Цей вектор подається на Dense32.

Dense32 – це повнозв’язний шар з 32 нейронами.

Кожний нейрон отримує всі 64 входи.

Матриця ваг має розмір:

$$W_2 \in \mathbb{R}^{32 \times 64}.$$

Тобто, 32 рядки (по одному для кожного нейрона), 64 стовпці (по одному для кожного входу).

$$W_2 = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,64} \\ w_{2,1} & w_{2,2} & \dots & w_{2,64} \\ \dots & \dots & \dots & \dots \\ w_{32,1} & w_{32,2} & \dots & w_{32,64} \end{bmatrix}.$$

Вектор bias визначається як:

$$b_2 = [b_2^{(1)}, b_2^{(2)}, \dots, b_2^{(32)}].$$

Шар обчислюється за формулою:

$$z_2 = W_2 h_1 + b_2,$$

Обчислення для нейрона 1:

$$z_2^{(1)} = w_{1,1} h_1^{(1)} + w_{1,2} h_1^{(2)} + \dots + w_{1,64} h_1^{(64)} + b_2^1.$$

4.5. Друга ReLU.

$$h_2 = \text{ReLU}(z_2),$$

$$h_2 \in \mathbb{R}^{32}.$$

4.6. Вихідний Dense шар (Dense1).

Також виконується лінійне перетворення:

$$z_3 = W_3 h_2 + b_3,$$

де

$$W_3 \in \mathbb{R}^{1 \times 64},$$

$$b_3 \in \mathbb{R},$$

$$z_3 \in \mathbb{R}.$$

Dense1 – це повнозв'язний шар з одним нейроном, тобто 32 входи і 1 вихід.

Матриця ваг має розмір:

$$W_3 \in \mathbb{R}^{1 \times 32},$$

$$W_3 = [w_1, w_2, \dots, w_{32}].$$

Таким чином, вихід обчислюється як:

$$z_3 = W_3 h_2 + b_3 = w_1 h_2^{(1)} + w_2 h_2^{(2)} + \dots + w_{32} h_2^{(32)} + b_3.$$

4.7. Лінійний вихід.

Виконується наступне:

$$y = z_3 = W_3 h_2 + b_3.$$

Це важливо для LLR корекції, бо значення можуть бути як додатні, так і від'ємні.

4.8. Повна математична модель MLP:

$$y = W_3 \text{ReLU}(W_2 \text{ReLU}(W_1 x + b_1) + b_2) + b_3.$$

5. Обчислення апостеріорного LLR після корекції:

$$L(u_k) = L_a(u_k) + \tilde{L}_e(u_k), \quad (8)$$

де $L_a(u_k)$ – апіорне LLR (від попереднього декодера); $\tilde{L}_e(u_k)$ – скориговане зовнішнє LLR від MLP.

6. Обмін зовнішніми LLR у турбо декодері.

1) Використовуємо перемешувач (interleaver) або деперемешувач (deinterleaver), щоб передати LLR іншому декодеру турбо коду:

$$L_a^{next}(u_k) = \text{interleaver}(L(u_k)). \quad (9)$$

2) Повторюємо кроки 2–6 для N ітерацій турбо декодування.

7. Підготовка MLP до навчання.

Застосуємо цільову функцію втрат:

$$\mathcal{L} = - \sum_k [u_k \ln \sigma(\tilde{L}_e(u_k)) + (1 - u_k) \ln (1 - \sigma(\tilde{L}_e(u_k)))], \quad (10)$$

де $\sigma(x) = \frac{1}{1+e^{-x}}$ – сигмоїдна функція для переведення LLR у ймовірність.

Метою є скориговані LLR, які відображають правильну ймовірність успіху декодування біта.

Існують три події щодо прийняття рішень при декодуванні декодером d , $d \in \overline{1,2}$ ітерації декодування j , $j \in \overline{1,I}$ біта інформації:

1) Подія A_1 . Зміни знаку в значеннях $L_a^{d,j}(x_t^C)$ і $L_e^{d,j}(x_t^C)$ ітерації j не відбувається ($\text{sign}(L_a^{d,j}(x_t^C)) = \text{sign}(L_e^{d,j}(x_t^C))$), $L(x_t^C) \geq 0$. Виноситься однозначне рішення щодо того, що був переданий біт $x_t^C = 1$.

2) Подія A_2 . Зміни знаку в значеннях $L_a^{d,j}(x_t^C)$ і ітерації j не відбувається ($\text{sign}(L_a^{d,j}(x_t^C)) = \text{sign}(L_e^{d,j}(x_t^C))$), $L(x_t^C) < 0$. Виноситься однозначне рішення щодо того, що був переданий біт $x_t^C = -1$.

3) Подія A_3 . Знак апіорного значення $L_a^{d,j}(x_t^C)$ та знак апостеріорної інформації $L_e^{d,j}(x_t^C)$ ітерації j не дорівнюють нулю ($\text{sign}(L_a^{d,j}(x_t^C)) \neq \text{sign}(L_e^{d,j}(x_t^C))$). У цьому випадку можливі помилки декодування.

Показник якості для d , $d \in \overline{1,2}$ при ітерації j , $j \in \overline{1,I}$ розраховується як:

$$\sum_{d=1}^2 R^{d,j}(t+1) = R^{d,j}(t) + 1,$$

$$\text{якщо } \text{sign}(L_a^{d,j}(x_t^C)) \neq \text{sign}(L_e^{d,j}(x_t^C)), t \in \overline{1,N}.$$

Чим частіше значення показника невизначеності R зростає, тим більше ймовірність появи неправильно декодованих бітів, що негативно впливає на достовірність прийому інформації.

Сумарний показник невизначеності R_Σ визначається як сума показників невизначеності по всіх ітераціях декодування:

$$R_\Sigma = \sum_{j=1}^I R^{d,j}.$$

Для зручності розрахунків та адаптації зробимо нормалізацію показника невизначеності:

$$\tilde{R}_\Sigma = \frac{R_\Sigma}{B \cdot \tilde{N} \cdot I} = \frac{\sum_{j=1}^I R^{d,j}}{B \cdot \tilde{N} \cdot I},$$

де B – кількість блоків даних деякого вікна спостереження, \tilde{N} – змінний розмір блоку даних, I – кількість ітерацій декодування турбо коду.

При параметричній адаптації, виходячи з обчисленого показника невизначеності для декодера d , $d \in \overline{1,2}$ на ітерації декодування j , $j \in \overline{1,I}$, та враховуючи накопичені значення невизначеності по B оброблених блоках даних, виконується адаптивне визначення оптимального розміру діаграми станів відповідного РСЗК.

8. Повний покроковий алгоритм (рис. 4).

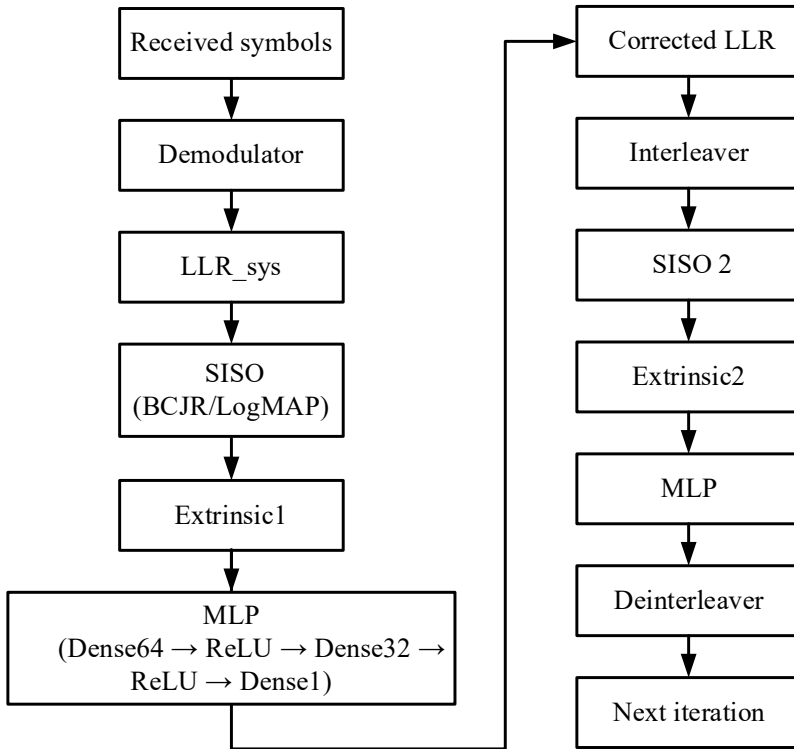


Рис. 4. Повний покроковий алгоритм

- 1) Ініціалізація: $L_a^{(0)} = 0$.
- 2) Для кожного біта u_k обчислити $L_e(u_k)$.
- 3) Створити feature vector f_k .
- 4) Передати f_k у MLP $\rightarrow \tilde{L}_e(u_k)$.
- 5) Обчислити апостеріорний LLR: $L(u_k) = L_a(u_k) + \tilde{L}_e(u_k)$.
- 6) Передати LLR іншому декодеру (interleaver).
- 7) Повторити N ітерацій.

Після останньої ітерації скориговані LLR використовуються для остаточного рішення (hard decision):

$$\hat{u}_k = \begin{cases} 1, & L(u_k) > 0 \\ 0, & L(u_k) < 0 \end{cases} \quad (11)$$

Висновки

1. Роботу присвячено дослідженню підвищення ефективності функціонування сучасних безпроводних технологій 5G та 6G.

2. В статті представлено метод параметричної адаптації паралельних та послідовних турбо кодів з використанням нейронних мереж типу багаторівневий перцепторон та показника невизначеності декодування.

3. Розглянуто використання нейронних мереж типу багаторівневий перцепторон для коригування зовнішніх логарифмічних відношень функцій правдоподібності ймовірнісних алгоритмів декодування паралельних та послідовних турбо кодів.

4. Запропоновано використання показника невизначеності декодування паралельних та послідовних турбо кодів на етапі навчання при визначенні вагових коефіцієнтів матриці ваг та при функціонуванні нейронних мереж.

СПИСОК ЛІТЕРАТУРИ / REFERENCES

1. Aldossari, S. A., Aldosary, A., & Chen, K.-C. (2023). Overcoming wireless channel modelling and relay signal selection via artificial intelligence techniques in 5G and beyond. In *Proceedings of the 14th International Conference on Ubiquitous and Future Networks (ICUFN)* (pp. 810–815). <https://doi.org/10.1109/ICUFN57995.2023.10200723>
2. Thamilchelvan, R., & Gomathy, C. (2023). Modeling intelligent wireless communication channel for 5G and beyond using advanced machine learning techniques. In *Proceedings of the 5th International Conference on Inventive Research in Computing Applications (ICIRCA)* (pp. 916–920). <https://doi.org/10.1109/ICIRCA57980.2023.10220931>
3. Al-Khafaji, M., & Elwiya, L. (2022). ML/AI empowered 5G and beyond networks. In *Proceedings of the International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)* (pp. 1–6). <https://doi.org/10.1109/HORA55278.2022.9799813>
4. Olaniyi, K. A., Heymann, R., & Swart, T. G. (2024). Machine learning for channel coding: A paradigm shift from FEC codes. *Journal of Communications*, 19(2), 107–118.
5. Richardson, T. J., Shokrollahi, M. A., & Urbanke, R. L. (2001). Design of capacity-approaching irregular low-density parity-check codes. *IEEE Transactions on Information Theory*, 47(2), 619–637. <https://doi.org/10.1109/18.910578>
6. Benedetto, S., & Montorsi, G. (1996). *Principles of turbo coding*. Springer.
7. Arikan, E. (2009). Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on Information Theory*, 55(7), 3051–3073. <https://doi.org/10.1109/TIT.2009.2021379>

Стаття надійшла до редакції 22.12.25, надійшла після рецензування 30.01.26, прийнята 24.02.26

The article was received 22.12.25, received after revision 30.01.26, accepted 24.02.26

Зайцева Лілія Ігорівна

аспірант Інституту телекомунікацій і глобального інформаційного простору Національної академії наук України

Адреса робоча: Україна, м. Київ, Чоколівський бульвар, 13

ORCID ID: <https://orcid.org/0000-0002-0668-711X> e-mail: lili5990n@ukr.net