

[0000-0002-2046-481X] **Е. В. Фауре**, *д-р техн. наук, професор*,
[0000-0002-3185-8427] **Б. А. Ступка**

e-mail: b.a.stupka.fitis20@chdtu.edu.ua

Черкаський державний технологічний університет
б-р Шевченка, 460, м. Черкаси, 18006, Україна

ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ ПРОЦЕСУ ВСТАНОВЛЕННЯ ЦИКЛОВОГО СИНХРОНІЗМУ В СИСТЕМАХ ЗВ'ЯЗКУ З НЕРОЗДІЛЬНИМ ФАКТОРІАЛЬНИМ КОДУВАННЯМ

У роботі досліджено ефективність методів циклової синхронізації систем передавання інформації з нероздільним факторіальним кодуванням: методу, що використовує як синхрокомбінацію перестановку, що має максимальне значення мінімальної відстані Хеммінга від її двійкового представлення до всіх її циклічних зсувів, та методу, який базується на використанні як синхрокомбінації перестановки чисел з її поділом на префіксну та суфіксну частини. Розроблено та описано структурні схеми імітаційних моделей системи передавання даних для кожного з методів. Описано середовище розробки та параметри апаратної частини, на якій виконувалося моделювання. Побудовано програмні імітаційні моделі систем передавання даних, у яких реалізовано алгоритми встановлення циклового синхронізму для кожного з наведених методів. Обґрунтовано основні модулі, які використовувалися як для реалізації моделей, так і для інтерпретації результатів. Оцінено показники ефективності методів циклової синхронізації: швидкість встановлення синхронізму та ймовірність хибного фазування залежно від ймовірності бітової помилки в каналі зв'язку. Продемонстровано особливості та переваги для кожного з методів.

Ключові слова: факторіальне кодування, циклова синхронізація, синхрокомбінація, ймовірність встановлення синхронізму, завада високої інтенсивності, імітаційне моделювання, короткий пакет.

Вступ. Процедура циклового фазування є невід'ємною частиною всіх стандартних протоколів мережевої взаємодії [1]–[5]. Одним із основних параметрів, за яким оцінюється якість системи циклової синхронізації, є час входження в синхронізм. Розвиток телекомунікаційних систем приводить до необхідності постійного зменшення часу входження в синхронізм як засобу зменшення витрат часу на процедуру встановлення з'єднання в загальному часі сеансу зв'язку. Ця обставина стимулює постійні пошуки нових технічних рішень, спрямованих на досягнення цієї мети.

Система циклової синхронізації після отримання потоку даних забезпечує ідентифікацію вхідних сигналів циклового вирівнювання (синхрокомбінації або синхрослова). Типовий формат кадру передбачає, що синхрокомбінація відправляється в кожному кадрі і використовується, крім кадрової синхронізації, для оцінки якості каналу.

Цю роботу спрямовано на розвиток теорії побудови телекомунікаційних систем з нероздільним факторіальним кодуванням [6]–[11]. Телекомунікаційні системи з нероздільним факторіальним кодуванням [8] використовують нестандартну та надлишкову струк-

туру кадру, в якій не передбачено окремого поля початку роздільника кадру (SFD). Циклова синхронізація базується на особливостях кодового слова факторіального коду з відновленням даних за перестановкою (ФКВД) [6].

У публікаціях [12], [13] розроблено та детально досліджено два методи входження в цикловий синхронізм для телекомунікаційних систем передавання інформації з нероздільним факторіальним кодуванням в умовах впливу в каналі зв'язку завад високої інтенсивності. Обидва методи використовують як синхрокомбінацію перестановку π з M символів. Кожен символ перестановки кодується рівномірним двійковим кодом з довжиною кодової комбінації $l_r = \lceil \log_2 M \rceil$ біт. Тоді довжина перестановки π , представленої в двійковому вигляді, дорівнює $n = l_r \cdot M$.

Перший метод [12] передбачає виконання таких кроків:

1) перший символ перестановки π записується послідовністю, що складається тільки з нулів, другий – тільки з одиниць; третій символ перестановки π в двійковому записі повинен починатися нулем, а останній символ – закінчуватися одиницею;

2) у накопичувач приймача записуються три фрагменти отриманої з каналу бітової послідовності. Довжина кожного з фрагментів дорівнює довжині перестановки – M символів або n біт;

3) за прийнятими фрагментами формується уточнена послідовність R , біти якої обчислюються за мажоритарним принципом на основі відповідних біт фрагментів;

4) в уточненій послідовності R перевіряється наявність комбінації $\underbrace{10\dots01\dots10}_{l_r}$;

5) уточнена послідовність R порівнюється з синхрокомбінацією π . У випадку їх збігів з точністю до одного символа процедура циклового фазування припиняється;

6) якщо комбінацію $\underbrace{10\dots01\dots10}_{l_r}$ в уточ-

неній послідовності R не знайдено або їх більше однієї, а також якщо суфікс уточненої послідовності R відрізняється від суфікса еталонної синхрокомбінації π більше, ніж на один символ, додатково накопичуються ще два фрагменти. Перехід на п. 3.

Після досягнення деякого заданого порогу кількість накопичених фрагментів не змінюється. Процес пошуку синхронізму триває, поки або не буде знайдений синхронізм, або не закінчиться ліміт часу на виконання пошуку синхронізму.

Другий метод [13] передбачає виконання таких кроків:

1) синхрокомбінацією є перестановка π довжини M , що має максимальне значення мінімальної відстані Хеммінга від її двійкового представлення до всіх її циклічних зсувів;

2) приймачем накопичуються прийняті з каналу зв'язку K блоків, що складаються з l фрагментів по n біт. Значення K і l змінюються відповідно до описаної в [13] методики;

3) для кожного блоку незалежно обчислюється уточнена послідовність R_k , $k \in [1, K]$. Кожний біт цієї послідовності обчислюється за мажоритарним принципом, враховуючи відповідні біти прийнятих фрагментів;

4) для кожної уточненої послідовності R_k обчислюються відстані Хеммінга до всіх циклічних зсувів синхрокомбінації π . Якщо для якогось із зсувів ця відстань не перевищує значення $d_{lim} = \lfloor (d-1)/2 \rfloor$, приймається рішення про відповідність прийнятої послідовності цьому зсуву;

5) якщо всі послідовності R_k , $k \in [1, K]$, відповідають одному і тому ж зсуву синхрокомбінації, синхронізм встановлено. В іншому випадку повторюються всі операції виявлення синхрокомбінації, починаючи з п. 2.

Кількість накопичених фрагментів може послідовно збільшуватися до деякого, заздалегідь заданого порогу.

Мета та задачі дослідження. Метою роботи є оцінка показників ефективності методів циклової синхронізації [12], [13] для телекомунікаційних систем передавання інформації з нероздільним факторіальним кодуванням на основі імітаційного моделювання та дослідження області застосування для кожного з методів.

Для досягнення поставленої мети необхідно виконати наступні задачі:

- побудувати імітаційну модель системи передавання інформації з нероздільним ФКВД;

- у блоках циклової синхронізації приймачів реалізувати алгоритми, побудовані за описаними методами;

- виконати дослідження часу входження в цикловий синхронізм для різних параметрів каналу зв'язку (значень бітової помилки в каналі зв'язку);

- виконати порівняльний аналіз отриманих показників часу входження в цикловий синхронізм;

- сформулювати рекомендації щодо використання методів циклової синхронізації для телекомунікаційних систем передавання інформації з нероздільним факторіальним кодуванням.

Виклад основного матеріалу. Виходячи з [13], перший метод [12] має наступні недоліки:

1) інтегральна функція часу входження в синхронізм не є оптимальною, оскільки запропонована процедура не враховує всю структурну надлишковість кодового слова ФКВД;

2) імовірність встановлення хибного синхронізму відносно велика, що особливо проявляється зі збільшенням імовірності бітової помилки в каналі зв'язку, а також для великих значень коефіцієнта накопичення.

За теоретичними оцінками з [13], другий метод має кращі кількісні показники встановлення циклового синхронізму телекомунікаційних систем передавання інформації з ФКВД в умовах впливу в каналі зв'язку завад високої інтенсивності. Зокрема, підвищується ймовірність встановлення циклового синхронізму для заданого коефіцієнта нако-

пичення, а також зменшується ймовірність встановлення хибного синхронізму.

Наведений на рисунку 1 роботи [13] графік свідчить про більш швидке досягнення відносною частотою встановлення вірного синхронізму заданого порогового значення для другого методу [13] порівняно з першим методом [12]. Разом з тим, існують діапазони значень кількості накопичених фрагментів, де

відносна частота встановлення вірного синхронізму для першого методу [12] вища. Водночас перший метод [12] не задовольняє поставленим вимогам щодо ймовірності встановлення хибного синхронізму.

Побудова та опис імітаційних моделей. Виконаємо побудову імітаційних моделей систем передавання даних (СПД) з ФКВД.

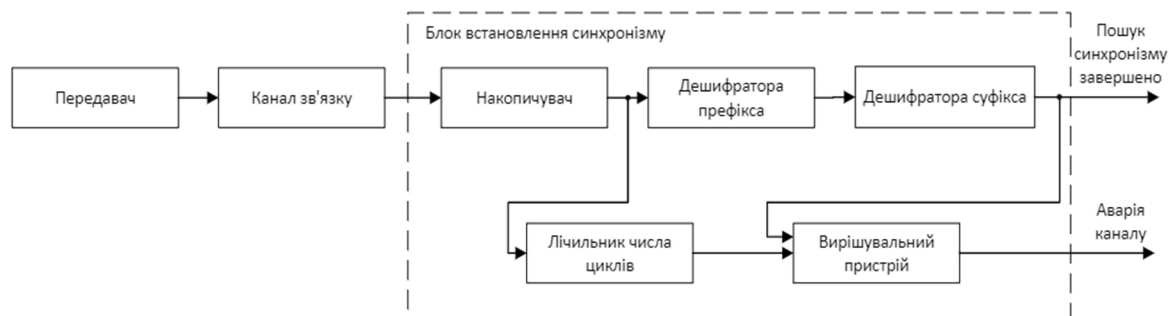


Рисунок 1 – Структурна схема імітаційної моделі СПД із застосуванням першого методу циклової синхронізації [12]

Розглянемо структурну схему імітаційної моделі СПД із застосуванням першого методу циклової синхронізації [12] (рисунку 1).

Для цього методу синхрокомбінація містить префікс, що виявляє межі символів. Прикладом такої синхрокомбінації для $M=8$ може бути перестановка

$$\pi = (000,111,001,010,100,110,101,011) [12].$$

Обрана послідовність записується в постійний запам'ятовувальний пристрій (ПЗП) передавача, який є джерелом синхрокомбінації передавальної станції.

У приймач надходить синхрокомбінація, уражена дією завад у каналі зв'язку і зсунута за фазою щодо положення ковзного вікна (виконаного у вигляді регістра зсуву). Водночас ні інтенсивність завади, ні зсув ковзного вікна (величина фазової неузгодженості циклів передавача та приймача) апріорно невідомі.

Під час старту процедури пошуку синхронізму встановлюється мінімальне значення коефіцієнта накопичення $l=3$. У буферний накопичувач записуються три фрагменти прийнятої з каналу зв'язку послідовності. Довжина кожного фрагмента дорівнює довжині синхрокомбінації. За цими трьома фрагментами мажоритарно обчислюється уточнена послідовність R .

У накопичувачі виконується побітовий циклічний зсув уточненої послідовності R .

Дешифратор префікса після виконання кожного зі зсувів перевіряє, чи виявлено префікс синхрокомбінації. Якщо префікс виявлено, то дешифратором суфікса перевіряється суфікс. Якщо обидві перевірки завершено позитивно, синхронізм встановлено. У цьому випадку дешифратор суфікса формує сигнал «Пошук синхронізму завершений». Він надсилається зворотним каналом на передавальну станцію.

Якщо ж дешифратором префікса не виявлено префікса синхрокомбінації або дешифратором суфікса не підтверджено цілісність синхрокомбінації, то в накопичувач додатково записуються ще два фрагменти отриманої з каналу послідовності біт. Після цього знову повторюються описані вище дії: обчислюється уточнена послідовність, за якою дешифратор префікса і дешифратор суфікса виконують виявлення синхрокомбінації. Такий пошук синхронного стану триває або до успішного завершення процедури пошуку, або до вичерпання ліміту часу на пошук синхронізму. Для цього система циклової синхронізації містить лічильник кількості циклів. Якщо після закінчення ліміту часу, відведеного на пошук синхронізму, він не знайдений, то вирішувальний пристрій формує сигнал «Аварія каналу». Процедура циклової синхронізації припиняється.

Розглянемо структурну схему імітаційної моделі СПД із застосуванням другого методу циклової синхронізації [13] (рисунку 2).

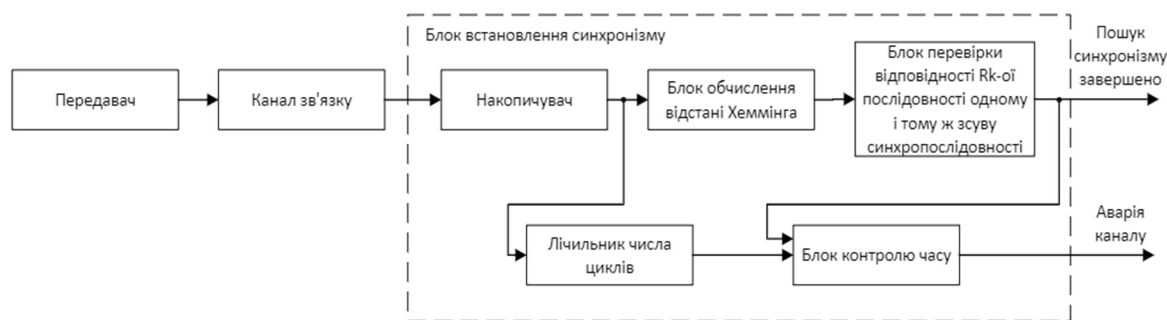


Рисунок 2 – Структурна схема імітаційної моделі СПД із застосуванням другого методу циклової синхронізації [13]

Синхрокомбінацією є перестановка π довжини M , що має максимальне значення мінімальної відстані Хеммінга від її двійкового представлення до всіх її циклічних зсувів. Для $M=8$ такою синхрокомбінацією може бути перестановка $\pi = (000, 001, 111, 011, 010, 101, 100, 110)$ [13]. Ця синхрокомбінація записується в ПЗП передавача, який стане джерелом синхрокомбінації передавальної станції.

У накопичувачі приймача накопичуються прийняті з каналу зв'язку K блоків, що містять l фрагментів по n біт. Значення K і l змінюються відповідно до методики [13], яка забезпечує дотримання заданої ймовірності хибного фазування. У накопичувачі також для кожного блоку незалежно обчислюється уточнена послідовність R_k , $k \in [1, K]$.

У блоці обчислення відстані Хеммінга для кожної уточненої послідовності R_k обчислюються відстані Хеммінга до всіх циклічних зсувів синхрокомбінації. Якщо для якогось із зсувів ця відстань не перевищує значення $d_{lim} = \lfloor (d-1)/2 \rfloor$, приймається рішення про відповідність прийнятої послідовності цьому зсуву.

У блоці перевірки відповідності R_k -ї послідовності одному і тому ж зсуву синхропослідовності виконується перевірка, чи всі послідовності R_k , $k \in [1, K]$, відповідають одному і тому ж зсуву синхропослідовності. Якщо це так, то системою циклової синхронізації приймається рішення про встановлення синхронізму. У цьому випадку блок перевірки на відповідність одному і тому ж зсуву синхрокомбінації для всіх R_k формує сигнал «Пошук синхронізму завершений». Він надсилається зворотним каналом на передавальну станцію.

В іншому випадку повторюються всі операції виявлення синхрокомбінації.

Як і у випадку з першим методом, пошук синхронного стану триває або до успішного завершення процедури пошуку, або до вичерпання ліміту часу на пошук синхронізму.

Опис програмного та апаратного забезпечення. Імітаційні моделі розроблено мовою програмування Python [14] з використанням інтегрованого середовища розробки PyCharm community edition 2020.3 [15].

Зазначимо, що PyCharm [15] створюється JetBrains на основі IntelliJ IDEA і є кросплатформним середовищем розробки, сумісним з Windows, macOS, Linux. Використання PyCharm [15] дозволило ефективно застосувати інструменти аналізу коду, графічний налагоджувач, запуск модульного тестування та підтримку веб-дозволів для Django.

Під час розробки моделі використовувалися наступні модулі Python:

- collections [16]–[19] – надає спеціалізовані типи даних на основі словників, кортежів, наборів, списків. З цього модуля в роботі використано collections. Counter – словник, який дозволяє підрахувати кількість незмінних об'єктів (в більшості випадків рядків). Модуль застосовано для полегшення роботи з результатами експериментів (процес, вихід);

- multiprocessing [17], [20], [21] – дає змогу створювати процеси так само, як і під час створення потоків за допомогою модуля потокової обробки. Цей модуль дозволяє обійти GIL (Global Interpreter Lock), що дає змогу лише одному потоку керувати інтерпретатором Python, і скористатися можливістю використання декількох (всіх) процесорів на комп'ютері. Модуль використано для пришвидшення виконання експериментів за рахунок розпаралелювання виконання експериментів на всіх можливих потоках використовуваного центрального процесора;

- random [14], [16] – для створення блоку даних із заданою ймовірністю бітової помилки p_0 ;

- os [14], [16] – для отримання від системи кількості потоків центрального процесора. Це значення використовується для модульної багатопроцесорної обробки.

Розробка програмної моделі відбувалася на персональному комп'ютері форм-фактора ноутбук з наступними параметрами:

- ОС – Windows 10;
- ЦП – Intel Core i5-8250U;
- ОЗУ 8gb (2x4gb dual channel 2133Mhz);
- Відеокарта – nVIDIA GeForce MX150 2gb;
- Сховище – INTEL SSDSC2KW256G8L (256 Гб, SATA-III).

Реалізація експериментів. Імітаційне моделювання мало на меті визначення таких параметрів:

- швидкість входження в синхронізм залежно від ймовірності бітової помилки p_0 . Цей показник отримано шляхом визначення середнього часу встановлення синхронізму для різних значень p_0 ;

- відносна частота хибного фазування залежно від ймовірності бітової помилки p_0 . Цей показник отримано шляхом визначення кількості хибних фазувань, поділеної на загальну кількість виконаних експериментів для конкретного значення p_0 .

Через значне збільшення часу виконання експерименту для великих значень ймовірності бітової помилки ($p_0 \geq 0.48$) кількість експериментів не була однаковою для всіх значень p_0 :

- для кожного значення $p_0 < 0.48$ проведено 10 000 експериментів;

- для кожного значення $p_0 \geq 0.48$ проведено 1000 експериментів.

Усі експерименти реалізовано в програмному середовищі PyCharm Community Edition [15] на персональному комп'ютері форм-фактора десктоп з наступними параметрами:

- ОС – Windows 10;
- ЦП – Intel Core i5-10400F;
- ОЗУ 32Gb (2x16Gb dual channel 3200Mhz);
- Відеокарта – GeForce GTX 1650 4Gb;
- Сховище – SSD M.2 2280 1TB Samsung.

Результати досліджень. Результати виконання експериментів та обробки їх результатів наведено в графічному вигляді.

Так, на рисунку 3 зображено графіки залежності середнього часу встановлення циклового синхронізму від ймовірності бітової помилки для кожного з розглянутих методів [12], [13].

На рисунку 4 зображено графіки залежності відносної частоти хибного фазування від ймовірності бітової помилки. Такі залежності дозволяють виконати оцінку надійності (достовірності) встановлення синхронізму для кожного з методів.



Рисунок 3 – Графіки залежностей середнього часу встановлення циклового синхронізму від ймовірності бітової помилки

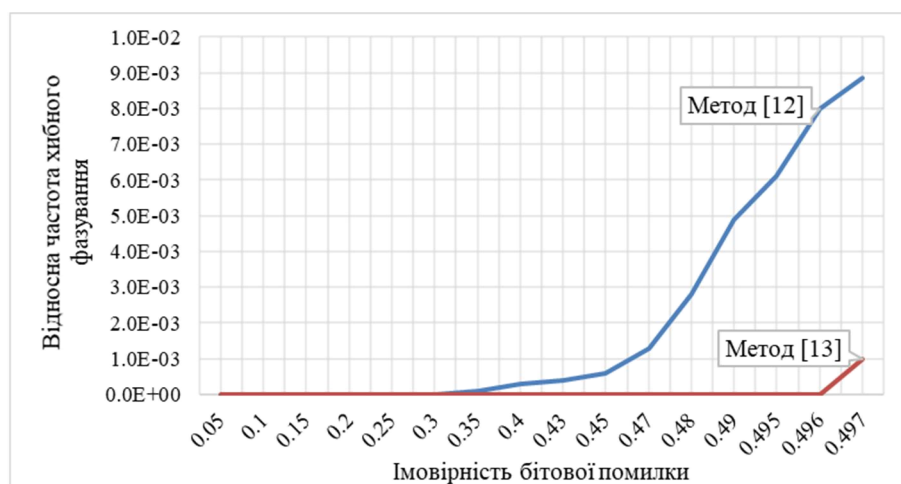


Рисунок 4 – Графіки залежностей відносної частоти встановлення хибного циклового синхронізму від імовірності бітової помилки

Обговорення результатів. Аналіз результатів експериментів свідчить про те, що:

- у каналах з рівнем бітової помилки $p_0 < 0.495$ швидкість встановлення циклового синхронізму є вищою для реалізації першого методу [12], для каналів з $p_0 \geq 0.495$ – для реалізації другого методу [12], [13];

- реалізація першого методу [12] має вищі значення ймовірності хибного фазування.

Зазначені результати пояснюються наступним:

- параметри в реалізації другого методу циклової синхронізації [13] не є оптимальними для всіх значень бітової помилки p_0 , а визначені для $p_0 = 0.495$. Для максимальної ефективності методу [13] імовірність бітової помилки в каналі має бути спрогнозована максимально точно. З другого боку, процедура вибору моментів зміни значень параметрів K і l може бути адаптивною та залежати від імовірності виникнення бітової помилки в каналі зв'язку;

- параметри другого методу циклової синхронізації [13] формуються, виходячи з вимог до забезпечення дотримання заданої ймовірності хибного фазування, дозволяючи визначити її на етапі проектування системи циклової синхронізації.

Висновки. У результаті представленого дослідження побудовано імітаційні моделі систем передавання інформації з нероздільним факторіальним кодуванням із застосуванням відомих методів циклової синхронізації ФКВД [12], [13]. Використання розроблених моделей дозволило виконати оцінку показників ефективності цих методів циклової синхронізації (часу входження в цикловий синхронізм, частоти

встановлення хибного циклового синхронізму для різних параметрів каналу зв'язку). Продемонстровано особливості та переваги для кожного з методів. Порівняльний аналіз отриманих показників дає змогу сформулювати рекомендації щодо застосування кожного з розглянутих методів.

Практична цінність роботи полягає в реалізації алгоритмів циклової синхронізації для ФКВД. У роботі детально описано середовище розробки та основні модулі, які використовувалися для реалізації алгоритмів.

Предметом подальших досліджень є побудова та дослідження системи циклової синхронізації [13] з адаптивним визначенням моментів зміни значень K і l у процесі її роботи, а також реалізація методів циклової синхронізації ФКВД на макетних зразках приймачів інформації СПД, аналіз показників їхньої ефективності та їх порівняння з теоретичними результатами та результатами моделювання.

Список використаних джерел

- [1] M. Popovic, *Communication protocol engineering*, 2nd ed. CRC Press, 2021.
- [2] D. Hercog, *Communication protocols: principles, methods and specifications*. Cham: Springer Nature Switzerland, 2020.
- [3] D. S. Dawoud, and P. Dawoud, *Serial communication protocols and standards: RS232/485, UART/USART, SPI, USB, INSTEON, Wi-Fi and WiMAX*. 2020.
- [4] A. Mannani, *Synthesis of communication protocols: a supervisory control approach*. S.I.: Scholars' Press, 2013.

- [5] H. König, *Protocol engineering*. Heidelberg ; New York: Springer, 2012.
- [6] Э. В. Фауре, "Факториальное кодирование с восстановлением данных", *Вісник Черкаського державного технологічного університету*, № 2, с. 33-39, 2016.
- [7] Е. В. Фауре, О. О. Харін, В. В. Швидкий, та А. І. Щерба, "Спосіб факторіального кодування з відновленням даних", 117004, Груд. 06, 2017.
- [8] J. S. Al-Azzeh, B. Ayyoub, E. Faure, V. Shvydkyi, O. Kharin, and A. Lavdanskyyi, "Telecommunication systems with multiple access based on data factorial coding", *Int. J. Commun. Antenna Propag.*, vol. 10, no. 2, pp. 102-113, 2020. doi: 10.15866/irecap.v10i2.17216.
- [9] E. V. Faure, "Factorial coding with error correction", *Radio Electron. Comput. Sci. Control*, vol. 0, no. 3, pp. 130-138, Nov. 2017. doi: 10.15588/1607-3274-2017-3-15.
- [10] E. V. Faure, V. V. Shvydkyi, A. O. Lavdanskyyi, and O. O. Kharin, "Methods of factorial coding of speech signals", *Radio Electron. Comput. Sci. Control*, vol. 0, no. 4, pp. 186-198. Nov. 2019. doi: 10.15588/1607-3274-2019-4-18.
- [11] E. V. Faure, A. I. Shcherba, and A. A. Kharin, "Factorial code with a given number of inversions", *Radio Electron. Comput. Sci. Control*, no. 2, pp. 143-153, 2018. doi: 10.15588/1607-3274-2018-2-16.
- [12] Е. В. Фауре, В. В. Швидкий, А. І. Щерба, О. О. Харін, та Б. А. Ступка, "Метод циклової синхронізації на основі перестановок", *Вісник Черкаського державного технологічного університету*, № 4, с. 67-76, 2020. doi: 10.24025/2306-4412.4.2020.222439.
- [13] E. Faure, A. Shcherba, and B. Stupka, "Permutation-based frame synchronisation method for short packet communication systems", in *2021 11th IEEE Int. Conf. on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Cracow, Poland, Sept. 2021, pp. 1073-1077. doi: 10.1109/IDAACS53288.2021.9660996.
- [14] "Python 3.9.6 documentation". [Online]. Accessed on: <https://docs.python.org/3>
- [15] *PyCharm: the Python IDE for Professional Developers by JetBrains*. JetBrains. [Online]. Available: <https://www.jetbrains.com/ru-ru/pycharm>.
- [16] Ch. Severance, *Python for Everybody: Exploring Data Using Python 3*. University of Michigan, 2016.
- [17] K. Liou, *Python Advanced Programming: The guide to learn python programming. Reference with exercises and samples about dynamical programming, multithreading, multiprocessing, debugging, testing and more*. Independently published, 2019.
- [18] W. McKinney, *Python for Data Analysis*, 2nd ed. O'Reilly Media, 2017.
- [19] J. VanderPlas, *Python Data Science Handbook*, 1st ed. O'Reilly Media, 2016.
- [20] Rick van Hattem, *Mastering Python: Master the art of writing beautiful and powerful Python by using all of the features that Python 3.5*. Packt Publishing, 2016.
- [21] G. Zaccane, *Python Parallel Programming Cookbook: Over 70 recipes to solve challenges in multithreading and distributed system with Python 3*, 2nd ed. Packt Publishing, 2019.

References

- [1] M. Popovic, *Communication protocol engineering*, 2nd ed. CRC Press, 2021.
- [2] D. Hercog, *Communication protocols: principles, methods and specifications*. Cham: Springer Nature Switzerland, 2020.
- [3] D. S. Dawoud, and P. Dawoud, *Serial communication protocols and standards: RS232/485, UART/USART, SPI, USB, INSTEON, Wi-Fi and WiMAX*. 2020.
- [4] A. Mannani, *Synthesis of communication protocols: a supervisory control approach*. S.I.: Scholars' Press, 2013.
- [5] H. König, *Protocol engineering*. Heidelberg ; New York: Springer, 2012.
- [6] E. Faure, "Factorial coding with data recovery", *Visnyk Cherkaskogo derzhavnogo tehnologichnogo universitetu*, no. 2, pp. 33-39, 2016 [in Russian].
- [7] E. V. Faure, O. O. Kharin, V. V. Shvydkyi, and A. I. Shcherba, "Method of factorial coding with data recovery", 117004, Dec. 06, 2017 [in Ukrainian].
- [8] J. S. Al-Azzeh, B. Ayyoub, E. Faure, V. Shvydkyi, O. Kharin, and A. Lavdanskyyi, "Telecommunication systems with multiple access based on data factorial coding", *Int. J. Commun. Antenna Propag.*, vol. 10, no. 2, pp. 102-113, 2020. doi: 10.15866/irecap.v10i2.17216.

- [9] E. V. Faure, "Factorial coding with error correction", *Radio Electron. Comput. Sci. Control*, vol. 0, no. 3, pp. 130-138, Nov. 2017. doi: 10.15588/1607-3274-2017-3-15.
- [10] E. V. Faure, V. V. Shvydkyi, A. O. Lavdanskyyi, and O. O. Kharin, "Methods of factorial coding of speech signals", *Radio Electron. Comput. Sci. Control*, vol. 0, no. 4, pp. 186-198, Nov. 2019. doi: 10.15588/1607-3274-2019-4-18.
- [11] E. V. Faure, A. I. Shcherba, and A. A. Kharin, "Factorial code with a given number of inversions", *Radio Electron. Comput. Sci. Control*, no. 2, pp. 143-153, 2018. doi: 10.15588/1607-3274-2018-2-16.
- [12] E. V. Faure, V. V. Shvydkyi, A. I. Shcherba, O. O. Kharin, and B. A. Stupka, "Method of cyclic synchronization based on permutations", *Visnyk Cherkaskogo derzhavnogo tehnologichnogo universitetu*, no. 4, pp. 67-76, 2020 [in Ukrainian]. doi: 10.24025/2306-4412.4.2020.222439.
- [13] E. Faure, A. Shcherba, and B. Stupka, "Permutation-based frame synchronisation method for short packet communication systems", in *2021 11th IEEE Int. Conf. on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Cracow, Poland, Sept. 2021, pp. 1073-1077. doi: 10.1109/IDAACS53288.2021.9660996.
- [14] "Python 3.9.6 documentation". [Online]. Available: <https://docs.python.org/3>.
- [15] *PyCharm: the Python IDE for Professional Developers by JetBrains*. JetBrains. [Online]. Available: <https://www.jetbrains.com/ru-ru/pycharm>.
- [16] Ch. Severance, *Python for Everybody: Exploring Data Using Python 3*, University of Michigan, 2016.
- [17] K. Lioy, *Python Advanced Programming: The guide to learn python programming. Reference with exercises and samples about dynamical programming, multithreading, multiprocessing, debugging, testing and more*, Independently published, 2019.
- [18] W. McKinney, *Python for Data Analysis*, 2nd ed. O'Reilly Media, 2017.
- [19] J. VanderPlas, *Python Data Science Handbook*, 1st ed. O'Reilly Media, 2016.
- [20] Rick van Hattem, *Mastering Python: Master the art of writing beautiful and powerful Python by using all of the features that Python 3.5*. Packt Publishing, 2016.
- [21] G. Zaccane, *Python Parallel Programming Cookbook: Over 70 recipes to solve challenges in multithreading and distributed system with Python 3*, 2nd ed. Packt Publishing, 2019.

E. V. Faure, Dr.Sc., Professor,

B. A. Stupka

e-mail: b.a.stupka.fitis20@chdtu.edu.ua

Cherkasy State Technological University

Shevchenko blvd, 460, Cherkasy, 18006, Ukraine

SIMULATION MODELING OF THE PROCESS OF ESTABLISHING FRAME SYNCHRONISM IN COMMUNICATION SYSTEMS WITH NON-SEPARABLE FACTORIAL CODING

The efficiency of frame synchronization for information transmission systems with non-separable factorial coding is investigated. The paper considers both the method that uses as a syncword a permutation with the maximum value of the minimum Hamming distance from its binary representation to all its cyclic shifts, and the method that uses as a syncword a permutation with its division into prefix and suffix parts.

Synchronization algorithms for each of the studied methods have been described. Structural diagrams for simulation models of data transmission system for each of the methods have been developed and described. The development environment and hardware parameters, on which the simulation has been performed, are described. Software simulation models of data transmission systems have been constructed. The algorithms for establishing frame synchronism for each of the above methods have been implemented. The main modules used both for the implementation of models and for the interpretation of results are substantiated. The use of the developed models allows to

evaluate the effectiveness of frame synchronization methods: the speed of synchronization and the probability of erroneous phasing depending on the bit error rate in communication channel.

The features and benefits for each of the methods are demonstrated based on the results of performance evaluations.

The practical value of the work consists in the implementation of the algorithms for establishing frame synchronization for factorial coding with data recovery by permutations.

The obtained research results can be useful not only in systems with non-separable factorial coding but also in classical data transmission systems with standard delimiter.

The subject of the future research is the construction and study of the frame synchronization system for the method with adaptive determining the moments for changing synchronization process parameters while the system is running, as well as implementation of the frame synchronization methods on prototypes of data receivers for information transmission system with factorial coding with data recovery by permutation.

Keywords: *factorial coding, frame synchronization, syncword, probability of establishing synchronism, high-intensity noise, simulation modeling, short packet.*