

СИНТЕЗ БАРКЕРОПОДІБНИХ КОДІВ З АДАПТАЦІЄЮ ДО ВЕЛИЧИНИ ЗАВАДИ

Олег Різник, Ольга М'яус

Національний університет "Львівська політехніка", м. Львів

oleh.y.riznyk@lpnu.ua, olha.m.miaus@lpnu.ua

<https://doi.org/10.37406/sXXIcp.2021.v2.8>

Вступ

У військовій галузі, де широко використовуються мобільні інтелектуальні роботи, безпілотні літальні апарати, мікросупутники, різноманітні мобільні транспортні системи, автоматизовані системи управління озброєнням важливою проблемою є організація надійного зв'язку між даними засобами та віддаленим центром керування з забезпеченням відповідного рівня криптографічного захисту та завадостійкості. Для розв'язання такої проблеми розробляються системи захисту та передачі даних з використанням баркероподібних кодів.

Розробка алгоритмів та методів для завадостійкого кодування з використанням шумоподібних кодів на основі баркероподібних послідовностей розглядалась у багатьох роботах [1, 3]. Алгоритми кодування та декодування в режимі реального часу з використанням послідовностей Баркера були розглянуті в [6]. Бездротові системи захисту, стиснення та передачі даних у режимі реального часу з визначеними параметрами описані в [11]. Алгоритм пошуку та виправлення завад на основі баркероподібних послідовностей показаний у [9]. У [5] представлена робота в режимі реального часу для одночасного радіолокаційного та просторового захищеного зв'язку.

Розроблення методів і засобів завадостійкого кодування з використанням шумоподібних кодів на основі баркероподібних послідовностей висвітлені в працях, зокрема: С. Мацуюкі, А. Цунеда, М. Кельман та Ф. Рівест – алгоритм кодування та декодування в реальному часі за допомогою послідовностей Баркера [2, 4]; А.М.Н. Алджалаї, К. Фенг, В.К.М. Леунг и Р. Уорд – безпроводні системи захисту, стиснення та передачі даних у реальному часі з заданими параметрами [1]; Е. В'єнно та Б. Байрам – алгоритм знаходження та виправлення завад на основі баркероподібних послідовностей [12]; П. Родрігес-Гарсія, Г. Ледфорд, К. Бейліс та Р.Дж. Маркс – робота в системах реального часу для одночасного радіолокаційного та просторово захищеного зв'язку [10].

Питання розроблення методів, алгоритмів і структур завадостійкого кодування-декодування даних та синтез на їх основі систем захисту даних у реальному часі висвітлені в працях, зокрема, М. Кельмана – методи виправлення завад із застосуванням кодів Баркера [4].

Проте у цих працях мало уваги надано застосуванню при розробці систем захисту та передачі даних у реальному часі сучасних швидких моделей та алгоритмів. Крім того, недостатньо розглянутими є питання поєднання в межах комплексних систем засобів завадостійкого кодування.

Розділ 1.

Баркероподібна кодова послідовність на основі ІКВ.

Аналітична модель системи – це перелік математичних рівнянь, які описують властивості баркероподібних кодів та дозволяють визначити властивість

завадостійкості коду.

Зокрема, послідовність Баркера задається таким математичним формулюванням

$$\{a_i\}_{i=1}^N \mid a_i \in \{+1, -1\} \quad (1)$$

Загальний коефіцієнт автокореляції функції визначається як

$$c_j = \sum_{i=1}^{N-j} a_i a_{i+j} \quad (2)$$

Автокореляційна функція задається виразом:

$$R(j) = \sum_{i=1}^{N-j} a[i] a^*[i+j] \quad (3)$$

А рівень головної пелюстки визначається через вираз:

$$ML = \left| \sum_{i=1}^{N-j} a_i a_i \right| = N \quad (4)$$

Для спрощення та покращання результатів вирішення задачі актуальним є вироблення нового підходу до синтезу кодів з добрими автокореляційними властивостями, які наближаються до кодів Баркера. Один з цих підходів базується на синтезі баркероподібних кодових послідовностей за допомогою ідеальних кільцевих в'язанок.

Побудова баркероподібних кодових послідовностей змінної довжини.

Розроблений метод побудови баркероподібних кодових послідовностей змінної довжини, заснований на використанні властивостей підкласу в'язанкових структур – багатократних ідеальних кільцевих в'язанок (ІКВ) [7, 8].

Багатократна ідеальна кільцева в'язанка представляє комбінаторну конфігурацію, яка утворена на послідовності N цілих чисел $K_N = (k_1, k_2, \dots, k_i, \dots, k_N)$, де всі суми $1, 2, \dots, S_N^R = S_N - 1$ з поруч розташованих чисел є значенням R різних кільцевих сум.

Відомо співвідношення, що об'єднує кількість чисел N , кратність R та суму S_N всіх чисел R - кратної ідеальної кільцевої в'язанки:

$$S_N = \frac{N(N-1)}{R} + 1 \quad (5)$$

Для побудови баркероподібної кодової послідовності змінної довжини за допомогою ІКВ зі значеннями порядку N кратності R виділяємо рядок із S_N , де пронумеровані у зростаючому порядку елементи одновимірного масиву і які заповнені одиницями при умові що номери елементів співпадають з числами ІКВ. У всі інші елементи масиву, які залишилися незаповненими, заносимо "нулі". Отримана послідовність з одиниць і нулів буде S_N - розрядною баркероподібною кодовою послідовністю змінної довжини, циклічним зсувом якої можна отримати й решту дозволених комбінацій цієї послідовності. Наведемо приклад такої баркероподібною кодовою послідовності в таблиці кодових комбінацій, що утворена на основі ІКВ порядку $N=9$ та кратності $R=4$ (табл. 1):

$$1. 1. 1, 2, 2, 5, 1, 3, 3 \quad (6)$$

Будь-яка з цих $S_N(S_N - 1) / 2$ різних пар кодових комбінацій послідовності містить точно R із N одиничних значень в однойменних розрядах, що відповідає властивостям ІКВ. Інші $N - R$ елементів однієї і циклічно зсунутої іншої

баркероподібної кодової послідовності відрізняються від елементів, що розташовані в однойменних розрядах послідовності.

Тоді мінімальна кодова відстань даної баркероподібної кодової послідовності відповідає формулі:

$$d_{\min} = 2(N - R) \quad (7)$$

Кількість помилок, які виявляються t_1 , і кількість помилок, які виправляються t_2 за допомогою баркероподібної кодової послідовності, визначаються на основі мінімальної кодової відстані:

$$t_1 \leq 2(N - R) - 1, t_2 \leq N - R - 1. \quad (8)$$

Таблиця 1

Баркероподібна кодова послідовність на основі ІКВ з $N=9$ та $R=4$

1	1	1	1	0	1	0	1	0	0	0	0	1	1	0	0	1	0	0
0	1	1	1	1	0	1	0	1	0	0	0	0	1	1	0	0	1	0
0	0	1	1	1	1	0	1	0	1	0	0	0	0	1	1	0	0	1
1	0	0	1	1	1	1	0	1	0	1	0	0	0	0	1	1	0	0
0	1	0	0	1	1	1	1	0	1	0	1	0	0	0	0	1	1	0
0	0	1	0	0	1	1	1	1	0	1	0	1	0	0	0	0	1	1
1	0	0	1	0	0	1	1	1	1	0	1	0	1	0	0	0	0	1
1	1	0	0	1	0	0	1	1	1	1	0	1	0	1	0	0	0	0
0	1	1	0	0	1	0	0	1	1	1	1	0	1	0	1	0	0	0
0	0	1	1	0	0	1	0	0	1	1	1	1	0	1	0	1	0	0
0	0	0	1	1	0	0	1	0	0	1	1	1	1	0	1	0	1	0
0	0	0	0	1	1	0	0	1	0	0	1	1	1	1	0	1	0	1
1	0	0	0	0	1	1	0	0	1	0	0	1	1	1	1	0	1	0
0	1	0	0	0	0	1	1	0	0	1	0	0	1	1	1	1	0	1
1	0	1	0	0	0	0	1	1	0	0	1	0	0	1	1	1	1	0
0	1	0	1	0	0	0	0	1	1	0	0	1	0	0	1	1	1	1
1	0	1	0	1	0	0	0	0	1	1	0	0	1	0	0	1	1	1
1	1	0	1	0	1	0	0	0	0	1	1	0	0	1	0	0	1	1
1	1	1	0	1	0	1	0	0	0	0	1	1	0	0	1	0	0	1

Наведемо формули, що визначають кількість помилок, які можуть бути виявлені t_1 та кількість помилок, які можуть бути виправлені t_2 баркероподібною кодовою послідовністю:

$$t_1 \leq 2(N - R) - 1, t_2 \leq N - R - 1. \quad (9)$$

Кодову відстань можна визначати через параметри ІКВ:

$$d_{\min} = S_N - 2(N - R) \quad (10)$$

Представимо формули визначення кількості помилок, які можуть бути виявлені чи виправлені на основі баркероподібної кодової послідовності:

$$\left. \begin{matrix} t_1 \leq 2(N - R) - 1 \\ t_2 \leq N - R - 1 \end{matrix} \right\}, \text{ при } S_N \geq 4(N - R) \quad (11);$$

$$\left. \begin{matrix} t_1 \leq S_N - 2(N - R) - 1 \\ t_2 \leq \frac{S_N - 2(N - R + 1)}{2} \end{matrix} \right\}, \text{ при } S_N < 4(N - R) \quad (12);$$

У наведених випадках значення параметрів N і R не пов'язані між собою якоюсь залежністю і вибираються довільно. В такому випадку виникає задача про встановлення найкращого співвідношення між N і R , за умови якого розглянута

баркероподібна кодова послідовність набуває додаткових властивостей щодо завадостійкості. Завадостійкість баркероподібної кодової послідовності збільшується зі зростанням різниці $P = N - R$.

Найбільше значення різниці P досягається за виконання умови:

$$S_N = 2N \tag{13}$$

Співвідношення між вибраними параметрами N і R , коли баркероподібна кодова послідовність здатна виявляти та виправляти найбільшу кількість помилок:

$$P = \begin{cases} N/2, & N - \text{парне} \\ (N-1)/2, & N - \text{непарне} \end{cases} \tag{14}$$

Побудовані на основі ІКВ баркероподібні кодові послідовності змінної довжини дають змогу виявляти до $N-1$ помилок, що відповідає 50% від довжини кодової послідовності або виправляти до $N/2-1$ помилок, що відповідає 25% від довжини кодової послідовності для парних, і виявляти до N , що відповідає 50% від довжини кодової послідовності або виправляти до $(N-1)/2$ помилок, що відповідає 25% від довжини кодової послідовності для непарних значень N .

Розділ 2. Програмний продукт призначений для синтезу баркероподібних послідовностей

Розроблений програмний продукт призначений для автоматичного пошуку баркероподібних послідовностей (кодів) (БпК) визначеного користувачем порядку.

Концептуальна модель даного засобу складається з трьох основних компонент системи:

- Інтерфейс користувача;
- Менеджер потоків;
- Алгоритмічне ядро пошуку баркероподібних кодів.

На рис. 1 відображено концептуальну модель програмного продукту з його компонентами та описом взаємодії.

Методика побудови баркероподібних кодових послідовностей на основі ідеальних кільцевих в'язанок. Методика побудови баркероподібних кодових послідовностей змінної довжини на основі ІКВ за критерієм мінімуму автокореляційної функції полягає в наступному:

- вибрати необхідний варіант ІКВ за довжиною L_N баркероподібної кодової послідовності, який відповідає сумі елементів ІКВ S_N порядку N кратності R [1];

- побудувати L_N - позиційний код $\mu_i, i=1,2,\dots,L_N$ з мінімальною автокореляційною функцією на основі вибраного варіанту ІКВ $(k_1, k_2, \dots, k_l, \dots, k_N)$, де на N позиціях кодової послідовності з номерами елементів $x_l, l=1,2,\dots,N$, які

обчислюються з формули $x_l \equiv 1 + \sum_{i=1}^l k_i \pmod{L_N}$, розмістити значення "1", а на решті $L_N - N$ позиціях кодової послідовності - значення "0".

Вибір, розробка методів та алгоритмів функціонування програмного засобу.

Програмний продукт реалізований за допомогою мови програмування C та деяких бібліотек для отримання додаткових функцій, як-от розподілення часу виконання за допомогою паралельних потоків.

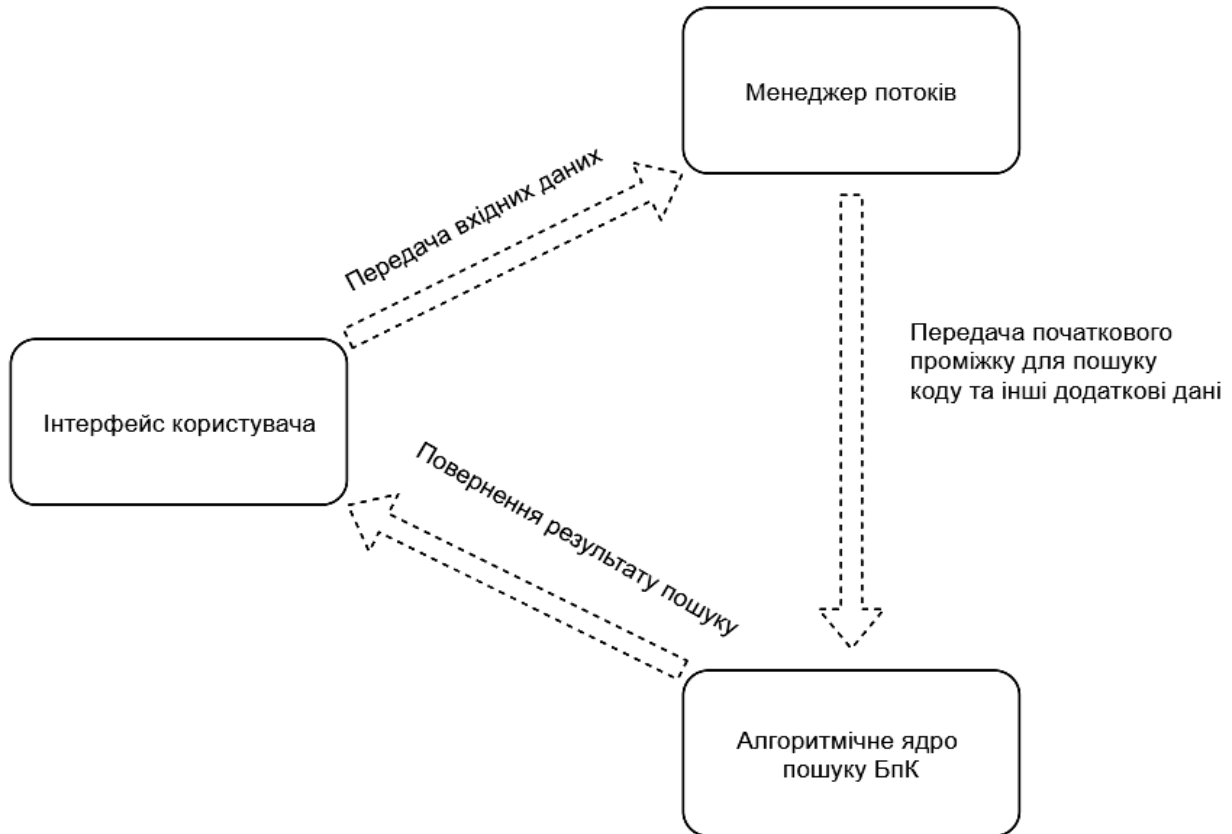


Рис. 1. Концептуальна модель системи

Обмеження. Загалом, програмний продукт виконує пошук баркероподібних кодів довжиною до 64. Таке обмеження має дві причини:

Для пришвидшення роботи з обчисленнями, алгоритм роботи програми працює з бітовими полями чисел. Таким чином, замість використання масивів даних, програма зберігає коди у одній змінній.

Швидкість пошуку кодів з порядком 30 і вище є досить повільним процесом, складність якого росте геометрично ($\times 2,1$). Для спрощення такої роботи, необхідно обмежити проміжок доступних довжин масиву.

Порядок дій. Програма виконує перебір кодів заданого порядку в певному проміжку чисел. Для спрощення пошуку, та використовуючи емпіричну властивість баркероподібних кодів, ми перевіряємо лише одну половину діапазону, оскільки у другій знаходяться такі ж самі, але обернені побітово елементи.

Також, для прискорення пошуку баркероподібних кодів, було здійснено ряд оптимізацій коду та перенесення його інфраструктури у багатопоточне середовище.

Загальна структура програмного продукту містить три головні компоненти, кожен з яких має також додатковий пов'язаний функціонал. Загальна діаграма файлової структури виглядає наступним чином:

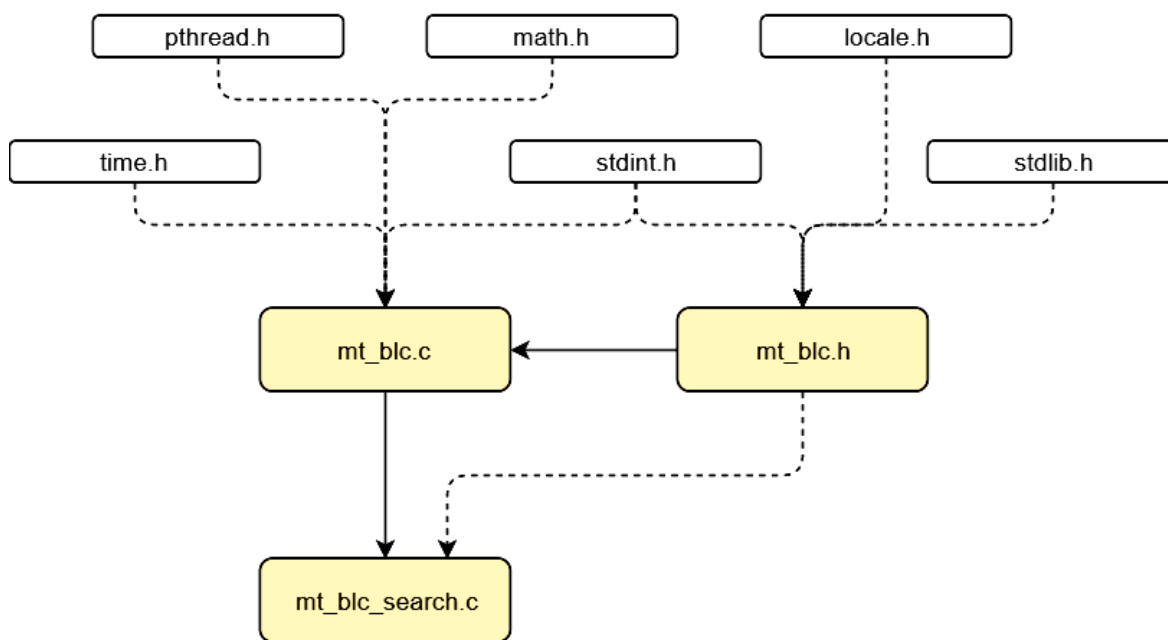


Рис. 2. Структурна схема програмного продукту

Алгоритм роботи програмного продукту базується на прямому пошуку баркероподібних кодів за допомогою функцій перевірки та збереження значення коду. Загальний алгоритм програми представлений на рис. 3.

Із блок-схеми алгоритму бачимо, що робота програми розпочинається з перевірки правильності введених користувачем параметрів. Ця процедура необхідна для зупинки виконання програми у разі помилкових вхідних даних, що дозволить зберегти користувача від витрат часу та вирішення помилок у ході виконання програми.

Наступним етапом є ініціалізація м'ютекса. М'ютекси - це об'єкти, за допомогою яких відбувається синхронізація доступу декількох потоків програми до спільного ресурсу. Синхронізація є важливим етапом роботи програми, оскільки у випадку її відсутності, неупорядкований запис даних про знайдені баркероподібні коди може призвести до багаторазового перезапису результатів, їх пошкодження чи аварійної зупинки програми.

Після ініціалізації, програма створює інші потоки виконання, які займатимуться перебором всіх кодів у заданому проміжку. Межі проміжку задаються у параметрах ініціалізації потоку головним потоком програми, а потім передаються у кожен окремий потік де використовуються для індексування по заданому проміжку кодів. Кожен потік, створений головним процесом, виконує пошук найкращих баркероподібних кодів за методом послідовного перебору. Проміжок кодів для виконання перебору визначається параметрами, які були передані цьому потоку під час його створення. Кожен варіант кодової послідовності, що дає найкращий результат, зберігається до місцевого буфера, де він зберігається до моменту завершення перебору, або до знаходження кращого коду.

Після завершення перебору всіх кодів з певного набору, потік має повернути отримані результати в головний процес. Для цього використовується глобальний масив найкращих баркероподібних кодів, який доступний для всіх потоків одночасно. Для уникнення проблем із несинхронізованим записом до спільних

ділянок пам'яті, потоки, перед тим, як записати свої результати, намагаються зайняти м'ютекс. Той потік, якому вдалось його зайняти, отримує право на запис до загального масив найкращих кодів. Після завершення запису своїх результатів, потік звільняє м'ютекс для інших потоків і завершує своє виконання.

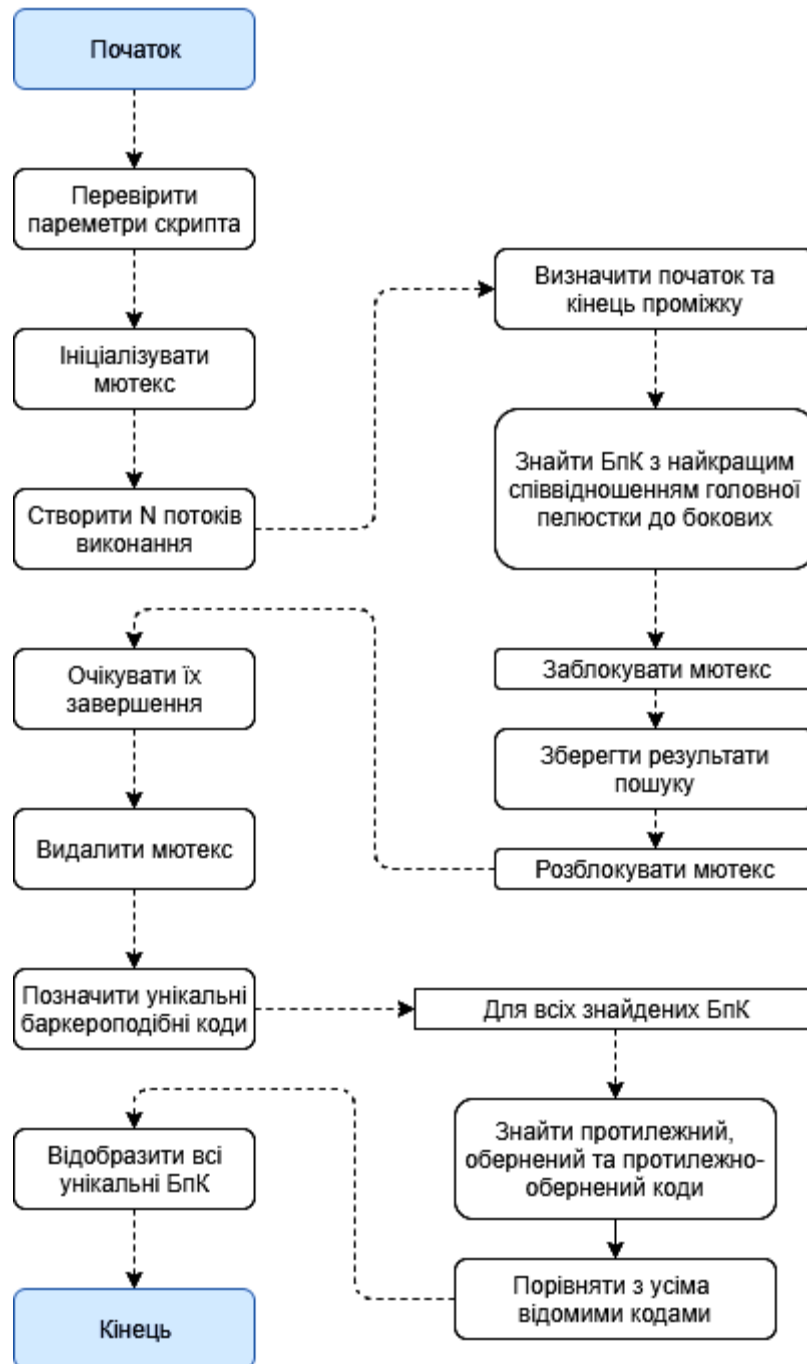


Рис. 3. Алгоритм роботи програми

Головний потік очікує на завершення роботи всіх потоків, після чого він видаляє м'ютекс і починає обробку результатів. У спільному масиві результатів, програма шукає унікальні кодові послідовності, відсіюючи повторення. Потім створюються три копії унікальних баркероподібних кодів, які є зміненими варіантами самих кодів.

Вони створюються за допомогою операцій відзеркалювання, інверсії та обох одночасно. Таким чином, програма отримує всеможливі варіації унікальних баркероподібних кодів, серед яких вона знову шукає повторення і відкидає їх.

Наприкінці виконання ми отримуємо набір абсолютно унікальних баркероподібних кодів, які програма виведе у зручному для перегляду виді та з усіма допоміжними даними.

Експериментальна частина.

Результатом виконання програми є перелік унікальних баркероподібних кодів з найбільшим показником надійності захисту даних. На рис. 4 відображено приклади результату виконання для 6, 13 та 14 порядкових кодів. «0» позначає -1, а 1 позначає 1.

```

Знайдено наступні UBBLС коди:
..... | ..... | ..... | ..... | ..... | ..... | ..... | ..11 1001
..... | ..... | ..... | ..... | ..... | ..... | ..... | ..11 1011
..... | ..... | ..... | ..... | ..... | ..... | ..... | ..11 1010
..... | ..... | ..... | ..... | ..... | ..... | ..... | ..11 1101
..... | ..... | ..... | ..... | ..... | ..... | ..... | ..11 0100
..... | ..... | ..... | ..... | ..... | ..... | ..... | ..11 0010
..... | ..... | ..... | ..... | ..... | ..... | ..... | ..10 1110
..... | ..... | ..... | ..... | ..... | ..... | ..... | ..10 0001

Знайдено 8 UBBLС коди з відхиленням 2 за 0.003386 сек [всього: 0.010113 сек].
oleksandr@creo2:~/projects/project_blc$ ./blc_search 13

Знайдено наступні UBBLС коди:
..... | ..... | ..... | ..... | ..... | ..... | ...1 1111 | 0011 0101

Знайдено 1 UBBLС коди з відхиленням 1 за 0.003699 сек [всього: 0.012287 сек].
oleksandr@creo2:~/projects/project_blc$ ./blc_search 14

Знайдено наступні UBBLС коди:
..... | ..... | ..... | ..... | ..... | ..... | ..11 1111 | 0011 0101
..... | ..... | ..... | ..... | ..... | ..... | ..11 1110 | 1011 0011
..... | ..... | ..... | ..... | ..... | ..... | ..11 1110 | 0110 1011
..... | ..... | ..... | ..... | ..... | ..... | ..11 1110 | 0110 1010
..... | ..... | ..... | ..... | ..... | ..... | ..11 1110 | 0110 0101
..... | ..... | ..... | ..... | ..... | ..... | ..11 1110 | 0010 1001
..... | ..... | ..... | ..... | ..... | ..... | ..11 1101 | 0110 0110
..... | ..... | ..... | ..... | ..... | ..... | ..11 1101 | 0100 1100
..... | ..... | ..... | ..... | ..... | ..... | ..11 1100 | 1101 0100
..... | ..... | ..... | ..... | ..... | ..... | ..11 1100 | 1100 1010
..... | ..... | ..... | ..... | ..... | ..... | ..11 1000 | 1010 0100
..... | ..... | ..... | ..... | ..... | ..... | ..11 0011 | 0000 0101
..... | ..... | ..... | ..... | ..... | ..... | ..11 0000 | 0100 1010
..... | ..... | ..... | ..... | ..... | ..... | ..10 1101 | 1111 0001
..... | ..... | ..... | ..... | ..... | ..... | ..10 1011 | 1110 0110
..... | ..... | ..... | ..... | ..... | ..... | ..10 1011 | 0011 1110
..... | ..... | ..... | ..... | ..... | ..... | ..10 1001 | 1000 0001
..... | ..... | ..... | ..... | ..... | ..... | ..10 1000 | 0001 1001

Знайдено 18 UBBLС коди з відхиленням 2 за 0.003383 сек [всього: 0.013639 сек].
    
```

Рис. 4. Вивід результату пошуку кодів для послідовностей довжиною 6, 13 та 14 чисел

З рис. 4 бачимо, що залежно від довжини необхідної кодової послідовності, ми отримуємо різну кількість найкращих кодових комбінацій. Отримані коди відсортовані у порядку спадання значення двійкового числа, а формат виведених даних дозволяє легко їх опрацювати та застосовувати у подальших дослідженнях.

Для подальшого аналізу результатів ми побудуємо графіки для кращого сприйняття статистичних даних. Графіки на рис. 5 та рис. 6 відображають зростання часу обчислень в залежності від кількістю елементів у шуканій послідовності. На рис.

5 графік має абсолютну вертикальну вісь, яка не завжди є зручною для загального аналізу. На рис. 6 графік базується на експоненційній шкалі – це дозволяє краще побачити співвідношення між різними розмірностями конфігурацій.

На рис. 7 відображено зміну швидкості роботи програми без та з використанням багатопоточності. Після переходу архітектури коду від однопоточного до багатопоточного типу, час роботи програми зменшився у чотири рази. Наслідком цього є швидший перебір кодових комбінацій та більше результату за той самий проміжок часу.

Залежність тривалості обчислень від кількості розрядів

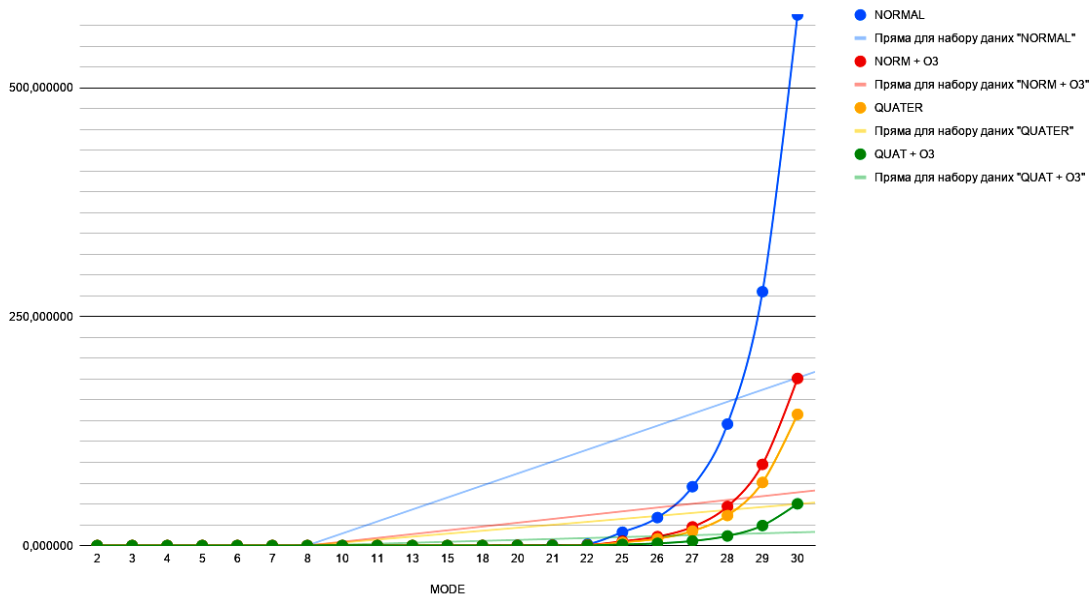


Рис. 5. Залежність тривалості пошуку від розміру послідовності

Залежність тривалості обчислень від кількості розрядів

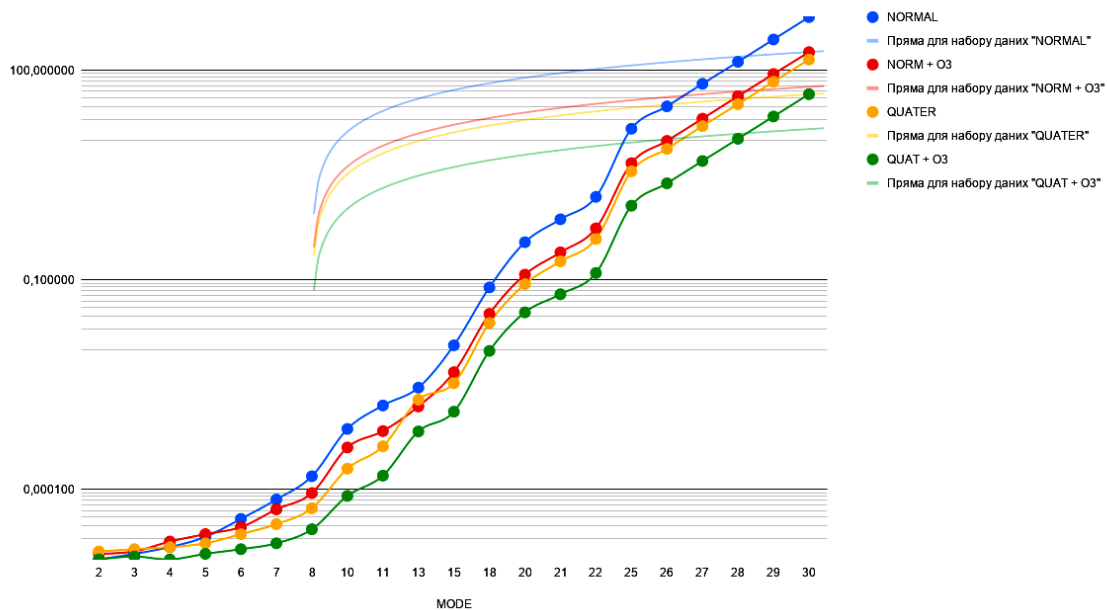


Рис. 6. Залежність тривалості пошуку від розміру послідовності у логарифмічній шкалі

Залежність тривалості обчислень від розрядності коду для послідовного та паралелізованого коду

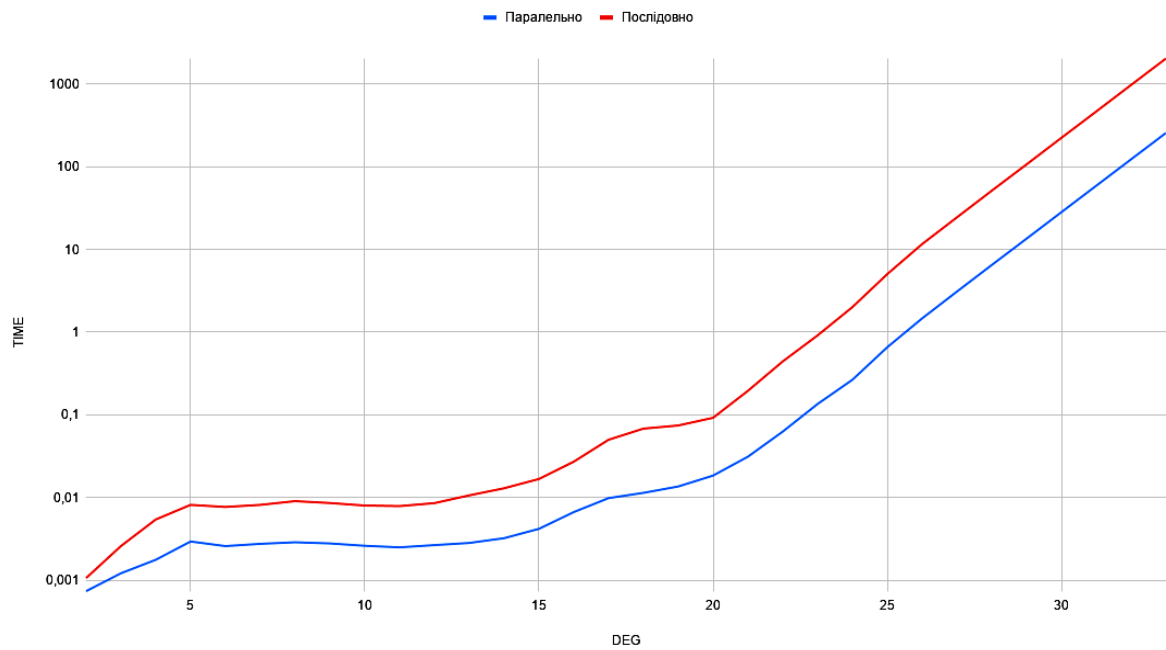


Рис. 7. Залежність тривалості послідовного та паралельного пошуку від розміру послідовності

На рис. 8 показано кількість знайдених унікальних елементів для кожного із ступенів масиву.

Кількість знайдених БпК для кожного ступеня

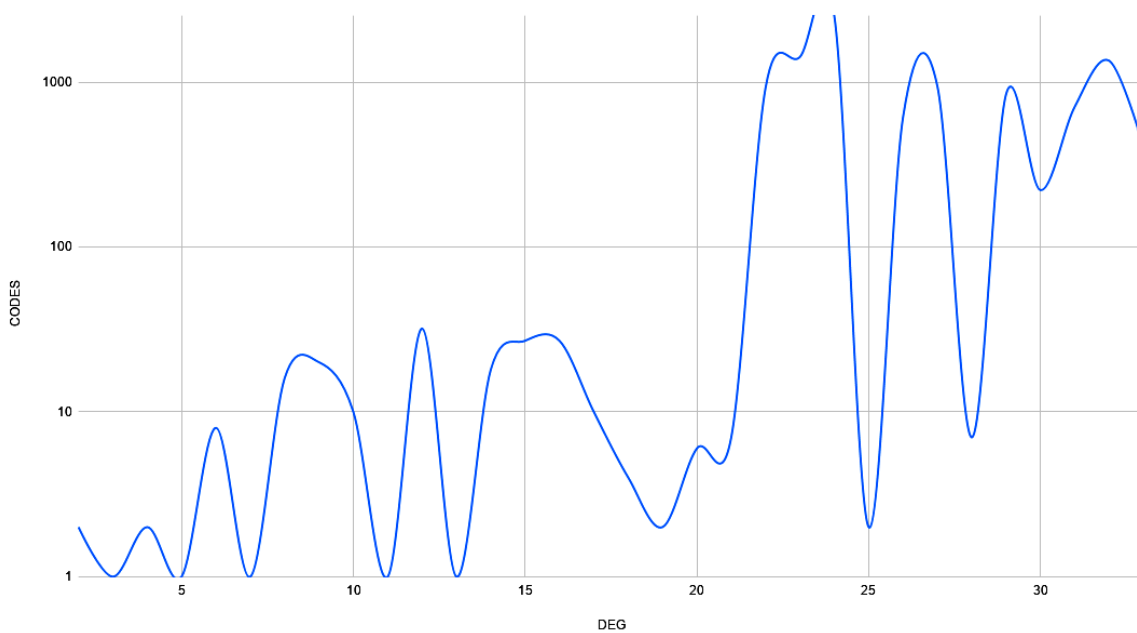


Рис. 8. Залежність кількості знайдених БпК від довжини послідовності

Тестування програмного продукту для синтезу баркероподібних послідовностей.

Для перевірки коректної роботи програми, були протестовані відомі існуючі результати баркероподібних кодів.

В результаті роботи програми співпали всі варіанти відомих існуючих баркероподібних кодів з результатами синтезу.

Результати тестування наведені в скріншоті на рис. 9, де:

N - довжина баркероподібного коду;

$g_{2,N}$ - кількість варіантів баркероподібних кодів;

$P_{i,j}$ - варіанти отриманих баркероподібних кодів.

N	$g_{2,N}$	$\{P_{i,j}\}$
2	1	1,-1
3	3	1,1,-1; 1,-1,1; 1,-1,-1
4	6	1,-1,-1,1; 1,1,-1,-1; 1,1,-1,1; 1,-1,1,1; 1,1,1,-1; 1,-1,-1,-1
5	12	1,1,1,-1,-1; 1,1,-1,-1,-1; 1,1,-1,1,1; 1,1,1,-1,1; 1,-1,-1,-1,1
6	14	1,-1,-1,-1,-1,1; 1,1,-1,1,-1,-1; 1,1,1,-1,1,1; 1,1,-1,1,1,1; 1,1,1,1,-1,1
7	20	1,-1,-1,-1,-1,-1,1; 1,1,1,1,-1,1,1; 1,1,-1,1,1,1,1; 1,1,-1,1,-1,1,1
8	32	1,-1,-1,1,-1,-1,-1,1; 1,-1,-1,-1,1,-1,-1,1; 1,1,1,-1,1,-1,1,1
9	40	1,-1,1,1,1,-1,-1,1; 1,-1,-1,1,1,1,-1,1; 1,1,1,-1,1,-1,1,1
10	20	1,1,1,1,-1,1,-1,-1,1,1; 1,1,-1,-1,1,-1,1,1,1; 1,1,1,1,-1,-1,1,1,-1,1
11	30	1,1,1,1,-1,1,-1,1,1,-1,-1; 1,1,-1,-1,1,-1,1,-1,-1,-1; 1,1,1,1,1,-1,1,-1,-1,1,1; 1,1,1,-1,1,-1,1,1,-1,1,1; 1,1,1,-1,1,-1,1,-1,-1,1,1
12	64	1,1,1,1,-1,-1,1,-1,1,1,1; 1,1,-1,1,-1,1,-1,-1,1,1,1; 1,1,1,1,-1,1,-1,1,-1,-1,1,1; 1,1,1,-1,1,-1,1,1,-1,-1,1,1; 1,1,-1,-1,1,-1,1,-1,1,1,1
13	46	1,1,1,1,-1,1,-1,1,1,-1,-1,1,1; 1,1,-1,-1,1,1,-1,1,1,1,1; 1,1,1,-1,1,1,1,-1,-1,1,-1,1,1; 1,1,-1,1,-1,-1,1,1,1,1,1
14	36	1,1,1,1,1,-1,-1,1,1,-1,1,1,1; 1,1,-1,1,-1,1,1,-1,-1,1,1,1,1; 1,1,-1,1,-1,1,1,-1,-1,1,1,1,1; 1,1,-1,-1,1,1,-1,1,-1,1,1,1,1
15	52	1,1,-1,1,-1,1,1,-1,-1,1,1,1,1,1; 1,1,1,1,1,1,-1,-1,1,1,-1,1,1,1; 1,1,1,1,-1,1,-1,1,-1,-1,1,1,-1,1,1; 1,1,1,1,-1,1,1,1,-1,-1,1,1,1,1
16	40	1,1,1,1,-1,-1,-1,1,-1,-1,1,1,-1,1,1; 1,1,1,1,-1,-1,1,1,-1,-1,1,-1,1,1,1; 1,1,-1,1,1,-1,1,-1,1,-1,-1,1,1,1,1
17	16	1,1,1,-1,-1,-1,-1,-1,-1,1,-1,-1,1,1,-1; 1,1,1,1,-1,-1,-1,-1,1,-1,-1,1,-1,-1,1,-1; 1,1,1,-1,1,1,1,-1,-1,-1,1,1,-1,1,-1
18	8	1,1,1,1,1,-1,1,-1,1,1,-1,1,1,1; 1,1,-1,-1,1,1,1,-1,1,-1,-1,1,-1,1,1,1,1; 1,1,1,1,-1,-1,1,1,-1,1,1,-1,1,1,-1,1,1
19	4	1,1,1,1,-1,-1,-1,1,1,-1,1,1,-1,1,1; 1,1,1,-1,-1,-1,1,-1,-1,1,-1,-1,1,-1,-1,1,-1,1; 1,-1,1,1,-1,1,1,-1,1,1,1,-1,-1,-1,1,1,1,1; 1,-1,1,-1,-1,1,-1,-1,1,-1,-1,-1,1,1,1,1
20	12	1,1,1,1,1,-1,1,1,-1,-1,1,-1,1,1,-1,-1,1; 1,-1,-1,-1,1,1,-1,1,-1,-1,1,1,-1,1,1,1,1
21	12	1,1,1,-1,-1,-1,-1,-1,1,-1,-1,1,-1,-1,1,-1,1; 1,1,-1,-1,-1,-1,1,1,1,-1,1,1,1,-1,1,1,-1,1

Рис. 9. Скріншот результатів тестування програми синтезу баркероподібних кодів

У першій колонці рис. 9 вказана кількість елементів в баркероподібному коді в порядку зростання, в другій - кількість отриманих при цьому баркероподібних кодів, в третій - приклади деяких баркероподібних кодів $P_{i,j}$, що ілюструють представлений раніше матеріал.

Таким чином, можна стверджувати, що розроблена програма працює коректно.

Висновки

Завадостійкість - це одна з важливих характеристик систем прийому/передачі даних. Підвищення завадостійкості при фіксованих швидкостях прийому/передачі

даних є актуальною проблемою, наприклад, для керування безпілотниками. Досліджені баркероподібні кодові послідовності дають змогу збільшити потужність отриманих послідовностей завдяки використанню дзеркальних завадостійких кодових послідовностей. Підвищення завадостійкості передачі даних досягається за рахунок збільшення довжини та потужності завадостійкої кодової послідовності, яка використовується для передачі одного повідомлення. Переваги цих послідовностей, такі як висока завадостійкість відповідно до вузькосмугових завад великої потужності, можливість поділу абонентів за кодовою ознакою, скритність передачі, висока стійкість до багатопробеневого поширення, висока роздільна здатність при навігаційних вимірах матимуть широке практичне використання у системах зв'язку та геолокації. Вдосконалено метод синтезу завадостійких баркероподібних кодових послідовностей з використанням ідеальних кільцевих в'язанок. Удосконалено метод для швидкого знаходження таких завадостійких кодових послідовностей, які здатні в найбільшому обсязі знаходити і виправляти помилки відповідно до довжини отриманої кодової послідовності. Реалізовано алгоритм для швидкого знаходження таких завадостійких баркероподібних кодових послідовностей, які здатні в найбільшому обсязі знаходити і виправляти помилки відповідно до довжини отриманої кодової послідовності. Розроблено імітаційну модель завадостійкого баркероподібного кодування з використанням ідеальних кільцевих в'язанок. Здійснено програмну реалізацію імітаційної моделі завадостійкого баркероподібного кодування щодо знаходження та виправлення помилок в отриманих завадостійких баркероподібних кодових послідовностях. Запропоновані завадостійкі баркероподібні кодові послідовності мають практичну цінність, оскільки за допомогою отриманої кодової послідовності досить просто та швидко знаходити до 50 % та виправляти до 25 % спотворених символів від довжини завадостійкої кодової послідовності.

Список використаних джерел

- [1] Aljalai A. M. N., Feng C., Leung V. C. M., Ward R. Improving the Energy Efficiency of DFT-s-OFDM in Uplink Massive MIMO with Barker Codes. *International Conference on Computing, Networking and Communications (ICNC)*. Big Island, HI, USA, 2020. pp. 731-735. DOI: 10.1109/ICNC47757.2020.9049829. <https://ieeexplore.ieee.org/document/9049829>
- [2] Kellman M., Rivest F., Pechacek A., Sohn L., Lustig M. Barker-Coded node-pore resistive pulse sensing with built-in coincidence correction. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. New Orleans, LA, 2017. pp. 1053-1057. DOI: 10.1109/ICASSP.2017.7952317. <https://ieeexplore.ieee.org/document/7952317>
- [3] Lakshmi C. R., Trivikramarao D., Subhani S., Ghali V. S. Barker coded thermal wave imaging for anomaly detection. *Conference on Signal Processing And Communication Engineering Systems (SPACES)*. Vijayawada, 2018. pp. 198-201. DOI: 10.1109/SPACES.2018.8316345.
- [4] Matsuyuki S., Tsuneda A. A Study on Aperiodic Auto-Correlation Properties of Concatenated Codes by Barker Sequences and NFSR Sequences. *International Conference on Information and Communication Technology Convergence (ICTC)*. Jeju, 2018. pp. 664-666. DOI: 10.1109/ICTC.2018.8539367.
- [5] Omar S. M., Kassem F., Mitri R., Hijazi H., Saleh M. A novel barker code algorithm for

- resolving range ambiguity in high PRF radars. *European Radar Conference (EuRAD)*. Paris, 2015. pp. 81-84. DOI: 10.1109/EuRAD.2015.7346242.
- [6] Pilsu Kim, Eunji Jung, Sua Bae, Kangsik Kim, Tai-kyong Song. Barker-sequence-modulated golay coded excitation technique for ultrasound imaging. *IEEE International Ultrasonics Symposium (IUS)*. Tours, 2016. pp. 1-4. DOI: 10.1109/ULTSYM.2016.7728737.
- [7] Riznyk O., Povshuk O., Kynash Y., Nazarkevich M., Yurchak I. Synthesis of non-equidistant location of sensors in sensor network. *XIV-th International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH)*. Lviv, 2018. pp. 204-208. DOI: 10.1109/MEMSTECH.2018.8365734.
- [8] Riznyk O., Povshuk O., Kynash Y., Yurchak I. Composing method of anti-interference codes based on non-equidistant structures. *XIIIth International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH)*. Lviv, 2017. pp. 15-17. DOI: 10.1109/MEMSTECH.2017.7937522.
- [9] Riznyk O., Povshuk O., Noga Y., Kynash Y. Transformation of Information Based on Noisy Codes. *IEEE Second International Conference on Data Stream Mining & Processing (DSMP)*. Lviv, 2018. pp. 162-165. DOI: 10.1109/DSMP.2018.8478509.
- [10] Rodriguez-Garcia P., Ledford G., Baylis C., Marks R. J. Real-Time Synthesis Approach for Simultaneous Radar and Spatially Secure Communications from a Common Phased Array. *IEEE Radio and Wireless Symposium (RWS)*. Orlando, 2019. pp. 1-4. DOI: 10.1109/RWS.2019.8714503.
- [11] Rosli1 S. J., Rahim H., Ngadiran R., Rani K. N. A., Ahmad M. I., Wee. Design of Binary Coded Pulse Trains with Good Autocorrelation Properties for Radar Communications. *MATEC Web of Conferences*. 2018. DOI: 10.1051/mateconf/201815006016.
- [12] Vienneau E., Byram B. Compound Barker-Coded Excitation for Increased Signal-to-Noise Ratio and Penetration Depth in Transcranial Ultrasound Imaging. *IEEE International Ultrasonics Symposium (IUS)*. Las Vegas, 2020. pp. 1-4. DOI: 10.1109/IUS46767.2020.9251650. <https://ieeexplore.ieee.org/document/9251650>