

**П. О. ВАСІН**

### АЛГОРИТМ ТЕХНОЛОГІЇ РОЗПІЗНАВАННЯ АКУСТИЧНИХ СИГНАЛІВ

Наведено алгоритм обробки відбитого акустичного сигналу для технології розпізнавання об'єктів на фоні шумових перешкод. Розглянута математична модель враховує, що вхідні дані (акустичні сигнали) мають раніше невідомий розподіл ймовірностей. Для побудови технології обробки акустичного сигналу використовувалися ймовірнісні моделі, побудовані на основі ланцюга Маркова, метод інтегрування Монте-Карло та побудований на ньому метод фільтра Bootstrap. На їх основі розроблено алгоритм обробки акустичного сигналу, відбитого від перешкод. Технологія обробки сигналів реалізує перехід від вхідних даних до представлення вхідних даних у вигляді сітки, де кожній комірці присвоєна характеристика-інтенсивність, яка наближається до нормального розподілу за центральною граничною теоремою та графічним методом. Для виявлення факту наявності перешкоди в довільній точці використана функція правдоподібності. Алгоритм готовий до практичного використання, що підтверджено прикладами. Графічно представлено результат роботи алгоритму розпізнавання двох рухомих об'єктів, які мають відбитий акустичний сигнал, близький до шумового. Відзначено залежність швидкості обробки від обсягу даних. Наведено декілька конструктивних підходів до вирішення цієї проблеми. Зауважено, що при використанні методів зберігання розріджених матриць досягається зменшення оцінки використаної пам'яті з квадратичної до лінійної при погіршенні оцінки швидкості у логарифм.

**Ключові слова:** акустичний сигнал, акустичний шум, технологія розпізнавання, ймовірнісні моделі, марківська модель простору-стану, метод інтегрування Монте-Карло, обчислювальне моделювання.

**П. А. ВАСИН**

### АЛГОРИТМ ТЕХНОЛОГИИ РАЗПОЗНАВАНИЯ АКУСТИЧЕСКИХ СИГНАЛОВ

Представлен алгоритм обработки отраженного акустического сигнала для технологии распознавания объектов на фоне шумовых помех. В рассмотренной математической модели учитывается, что входные данные (акустические сигналы) имеют заранее неизвестное распределение вероятности. Для построения технологии обработки акустического сигнала использованы вероятностные модели, построенные на основе цепи Маркова, метод интеграции Монте-Карло и построенный на нем метод фильтра Bootstrap. На их основе разработан алгоритм обработки акустического сигнала, отраженного от препятствий. В технологии обработки сигнала реализован переход от исходных данных к представлению входных данных в виде сетки, где каждой ячейке ставится в соответствие характеристика-интенсивность, которая приближается к нормальному распределению по центральной предельной теореме и графическому методу. Для выявления факта наличия помехи в случайной точке использована функция правдоподобия. Алгоритм готов для практического использования, что подтверждено примерами. Графически представлены результат работы алгоритма по распознаванию двух движущихся объектов, имеющих отраженный акустический сигнал, близкий к шуму. Отмечена зависимость скорости обработки от объема данных. Приведены несколько конструктивных подходов для решения данной проблемы. Замечено, что при использовании методов хранения разреженных матриц достигается уменьшение оценки использованной памяти с квадратической до линейной при ухудшении оценки скорости в логарифм.

**Ключевые слова:** акустический сигнал, акустический шум, технология распознавания, вероятностная модель, марковская модель пространства-состояния, метод интегрирования Монте-Карло, вычислительное моделирование.

**P. O. VASIN**

### ALGORITHM FOR RECOGNITION TECHNOLOGY OF ACOUSTIC SIGNALS

An algorithm for processing the reflected acoustic signal for the technology of object recognition against the background of noise interference is presented. The considered mathematical model takes into account that the input data (acoustic signals) have a previously unknown probability distribution. To build an acoustic signal processing technology the Markov chain based probabilistic models, the Monte Carlo integration method and the Bootstrap filter method built on it were used. Based on them, an algorithm for processing an acoustic signal reflected from obstacles was created. The signal processing technology implements the transition from the input data to the representation of the input data in the form of a grid, where a characteristic-intensity normally distributed according to the central limit theorem is assigned to each cell. A likelihood function is used to detect noise at a random point. The algorithm is ready for practical use, which is confirmed by examples. The result of the algorithm application for the recognition of two moving objects, which have a reflected acoustic signal close to the noise, is graphically presented. The dependence of the processing speed on the amount of data is noted. Several constructive approaches to solving this problem are given. It is noted that by using the methods of storing sparse matrices the estimate of the memory used is reduced from quadratic to linear with deterioration of the speed estimate by a logarithm.

**Key words:** acoustic signal, acoustic noise, recognition technology, probabilistic model, Markov space-state model, Monte Carlo integration method, computational modeling.

**Вступ.** *Сонар є дієвим інструментом для визначення координат розташування об'єктів та визначення їх геометричних характеристик. Методи та алгоритми обробки віддзеркаленого акустичного сигналу є основними засобами, які визначають якість гідроакустичної системи. Метою даної роботи є побудова алгоритму розпізнавання об'єктів методом обробки віддзеркаленого акустичного сигналу від перешкод на фоні шуму.*

Об'єктом для пошуку та розпізнавання може виступати перешкода у вигляді різних видів плаваючих засобів (човнів, катерів, кораблів тощо) або плавець. Шумами вважаються сигнали, які прийшли від випадкових джерел – *донних перешкод* (неоднорідностей дна, каміння), або помилкових сигналів від приладів тощо.

Треба зауважити, що більшість віддзеркалених сигналів від шуканих об'єктів виділяються амплітудою на фоні шуму. В таких випадках сигнали виділяються за допомогою *порогового методу*. Але в більш узагальнених випадках такий метод може дати помилку. Наприклад, якщо матеріал шуканого об'єкту здатен викривляти, поглинати сигнал або віддзеркалювати його в іншому напрямку. В результаті чого амплітуди віддзеркаленого сигналу будуть порівняні з амплітудою шумового сигналу, але при застосуванні тільки порогового методу неможливо гарантовано виявити об'єкт. Таким чином, методи виявлення, що базуються на амплітудних принципах обробки сигналів, не здатні забезпечити гарантоване визначення перешкоди.

Для розв'язання вищезазначеної проблеми в даній роботі пропонується технологія обробки сигналу по принципу багатокрокового фільтру з послідовною обробкою віддзеркаленого сигналу при застосуванні *методу Монте-Карло*, основні ідеї, принципи та приклади застосування якого представлено у роботах [1 – 5].

**Означення та теоретичні відомості для формулювання задачі.** Для простоти будемо використовувати  $x_t$  для позначення випадкової величини і її реалізації. Позначимо неперервну розподілену ймовірності за допомогою  $p(dx_t)$  замість  $\Pr(X_t \in dx_t)$  і дискретний розподіл за допомогою  $p(x_t)$  замість  $\Pr(X_t = x_t)$ .

При створенні технології обробки сигналів буде застосовано *нормальний розподіл (розподіл Гауса)* – розподіл ймовірностей випадкової величини, що характеризується густиною ймовірності:

$$N(x; \mu, \sigma) = f(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \quad (1)$$

де  $\mu$  – математичне сподівання;  $\sigma$  – дисперсія випадкової величини.

**Теорема 1 (Центральна гранична теорема).** Нехай  $\{X_k\}$  – послідовність взаємно незалежних випадкових величин з однаковими розподілами. Та існують скінченні: математичне сподівання  $\mu = E(X_k)$  та дисперсія  $\sigma^2 = D(X_k)$ . Нехай  $S_n = X_1 + \dots + X_n$ . Тоді

$$\sqrt{n}\left(\frac{S_n}{n} - \mu\right) \rightarrow N(0, \sigma^2).$$

**Означення: Функцією правдоподібності** будемо називати функцію  $\mathcal{L}(\theta | x)$ , яка показує правдоподібність набору значень параметрів  $\theta$  за заданих виходів  $x$  та дорівнює ймовірності цих спостережуваних виходів за заданих значень параметрів, тобто:  $\mathcal{L}(\theta | x) = P(x|\theta)$ .

**Математична постановка задачі.** Згідно з [5], обмежимося сигналами, що відповідають *Марківському процесу у нелінійній моделі простору-стану*. Не спостережуваний сигнал (приховані стани)  $\{x_t; t \in N\}$ ,  $x_t \in X$  моделюється як марківський процес з початковим розподілом  $p(x_0)$  і перехідним рівнянням  $p(x_t|x_{t-1})$ . Спостереження  $\{y_t; t \in N^*\}$ ,  $y_t \in Y$  приймаються умовно незалежними з урахуванням процесу  $\{x_t; t \in N\}$  та граничного розподілу  $p(y_t|x_t)$ . Підсумовуючи, модель процесу має такий опис:

$$\begin{array}{ccccccc} x_1 & \rightarrow & x_2 & \rightarrow & \dots & \rightarrow & x_m \\ \downarrow & & \downarrow & & & & \downarrow \\ y_1 & & y_2 & & & & y_m \end{array} \quad (2)$$

$$\begin{array}{l} p(x_0), \quad \text{при } t = 0; \\ p(x_t|x_{t-1}), \quad \text{при } t \geq 1; \\ p(y_t|x_t), \quad \text{при } t \geq 1. \end{array}$$

Тобто, сигнал і спостереження, згідно до часу  $t$ , позначаються через  $x_{0:t} \triangleq \{x_0, \dots, x_t\}$  і  $y_{1:t} \triangleq \{y_1, \dots, y_t\}$  відповідно.

**Мета задачі** полягає в оцінці *рекурсивної в часі апостеріорної ймовірності*  $p(x_{0:t}|y_{1:t})$ , пов'язаного з нею граничного розподілу величин  $p(x_t|y_{1:t})$ , та математичного очікування:

$$I(f_t) = E_{p(x_{0:t}|y_{1:t})} [f_t(x_{0:t})] \triangleq \int f_t(x_{0:t}) p(x_{0:t}|y_{1:t}) dx_{0:t}, \quad (3)$$

для деякої *функції інтересу*  $f_t: X^{t+1} \rightarrow R^{n_{f_t}}$ , що інтегрована по  $p(x_{0:t}|y_{1:t})$ . Приклади відповідної функції включають умовне середнє, в такому випадку  $f_t(x_{0:t}) = x_{0:t}$ .

Вважаємо, що у будь-який час  $t$ , апостеріорна ймовірність визначається за *теоремою Байєса*:

$$p(x_{0:t}|y_{1:t}) = \frac{p(y_{1:t}|x_{0:t})p(x_{0:t})}{\int p(y_{1:t}|x_{0:t})p(x_{0:t})dx_{0:t}}. \quad (4)$$

В результаті, отримуємо пряму рекурсивну формулу для даного спільного розподілу  $p(x_{0:t}|y_{1:t})$ :

$$p(x_{0:t+1}|y_{1:t+1}) = p(x_{0:t}|y_{1:t}) \frac{p(y_{t+1}|x_{t+1})p(x_{t+1}|x_t)}{p(y_{t+1}|y_{1:t})}. \quad (5)$$

Також граничний розподіл  $p(x_t|y_{1:t})$  задовольняє наступним рекурсіям:

– Прогнозування:  $p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1}$ ; (6)

– Оновлення:  $p(x_t|y_{1:t}) = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{\int p(y_t|x_t)p(x_t|y_{1:t-1})dx_t}$ . (7)

**Перетворення від множини точок до сітки та функція правдоподібності.** Вважаємо, що вхідні дані ма

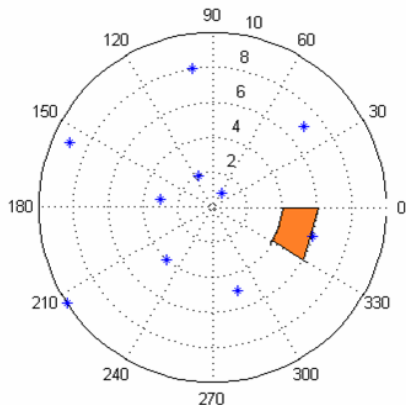


Рис. 1 – Представлення вхідних сигналів у системі полярних координат.

ють невідомий випадковий характер. Перетворимо вхідні дані таким чином, щоб вони відповідали відомому розподілу (саме нормальному розподілу). Для цього перейдемо до представлення вхідних даних у вигляді сітки (рис. 2). Кожна клітинка сітки буде мати свою характеристику – *інтенсивність клітинки*, яка буде залежати від вхідних даних.

Нехай  $n_{ij}$  – кількість сигналів, які потрапили в окіл точки  $p_{ij}$ ,  $\xi_k^{ij}$  – позиція сигналу,  $A_k^{ij}$  – амплітуда сигналу,  $\bar{A}^{ij}$  – середнє значення амплітуди сигналу, що, за законом великих чисел, прямує до математичного середнього при  $n \rightarrow \infty$ .

Введемо  $z_{ij}$  – інтенсивність клітинки  $(i, j)$ :

$$z_{ij} = \frac{1}{\sqrt{n_{ij}}} \sum_{k=1}^{n_{ij}} (A_k^{ij} - \bar{A}^{ij}) m_k^{(ij)}, \quad (8)$$

де за ваговий коефіцієнт  $m_k$  обирається:

$$m_k^{(ij)} = h_{ij}(\xi_k^{ij}) = e^{-\frac{a\xi_k^{ij} - p_{ij}^2}{m}}$$

$m$  – кількість клітинок;  $p_{ij}$  – позиція центру клітинки;  $a$  – коефіцієнт розмиття для експоненціальної функції.

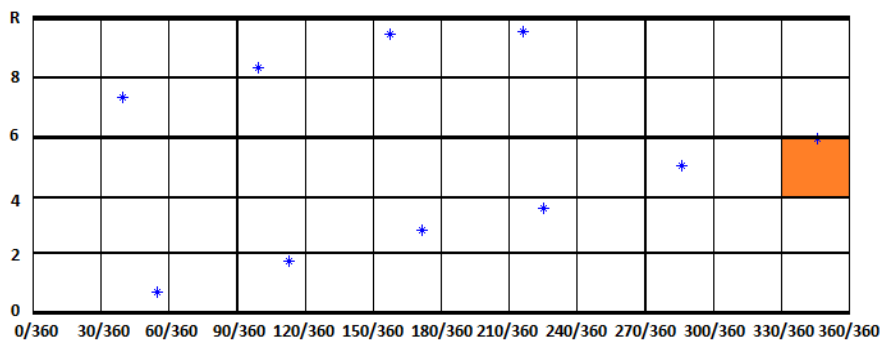


Рис. 2 – Представлення вхідних сигналів у вигляді сітки.

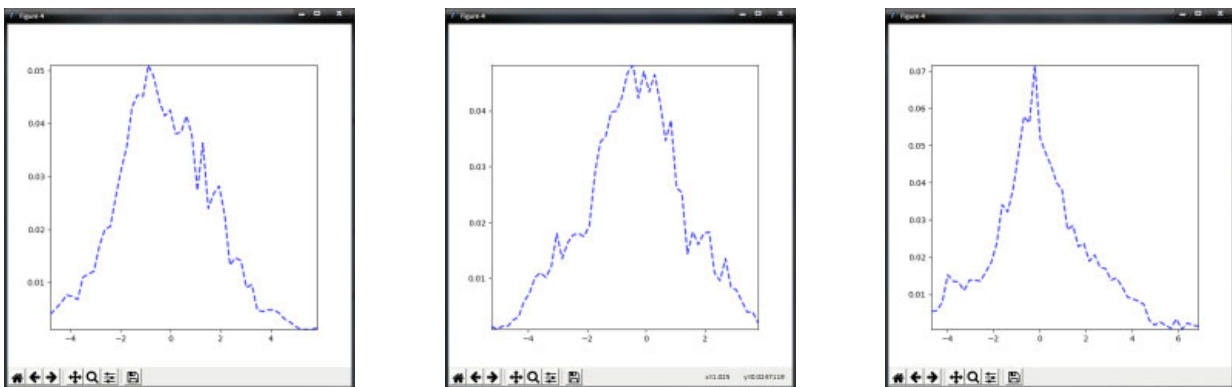


Рис 3 – Приклад функцій розподілу для  $z_{ij}$  на довільно взятих ітераціях, що наближаються до нормально розподіленої.

Дане представлення має властивості: ваговий вплив кожної точки (її амплітуда) розподіляється на всі клітинки в залежності від відстані до точки, яка відповідає клітинці. За центральною граничною теоремою, при достатньо великій кількості точок (вхідних даних) величина  $z_{ij}$  матиме нормальний розподіл. При наявності пе

решкоди в околі точки  $p_{ij}$  математичне сподівання величини  $z_{ij}$  буде відрізнятись від нуля тим більше, чим ближче знаходиться перешкода до точки  $p_{ij}$  або чим більше відрізняється амплітуда перешкоди від математичного сподівання амплітуди шуму. Графічне відтворення функції щільності величини  $z_{ij}$  на довільно взятих ітераціях для випадково згенерованого масиву сигналів демонструє наближення до нормального розподілу (рис. 3).

Для виявлення факту наявності перешкоди в довільній точці визначимо функцію правдоподібності у цій точці.

Нехай в точці  $p$  знаходиться об'єкт, віддзеркалений сигнал від якого має, в середньому, амплітуду рівну  $I_k$ . Тоді функція правдоподібності показує, з якою імовірністю при такій умові були отримані саме такі значення  $Z_k$  (бо оскільки на дані впливають різні види шуму, отримані дані мають випадковий характер).

Підкорегована для нормально розподіленого шуму формула, згідно [4], для функції правдоподібності буде мати вигляд:

$$\begin{aligned} \mathcal{L}(Z_k; I_k, p, \sigma) &= \frac{\text{вірогідність, що об'єкт присутній}}{\text{вірогідність, що об'єкту не присутній}} = \\ &= \prod_{ij} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z_{ij}^k - \mu_{ij})^2}{2\sigma^2}} = \prod_{ij} e^{-\frac{\mu_{ij}^2 - 2\mu_{ij}z_{ij}^k}{2\sigma^2}}, \end{aligned} \quad (9)$$

де  $\sigma$  – середнє квадратичне відхилення (дисперсія);  $\mu_{ij} = I_k h_{ij}(p)$ ;  $p$  – точка, в якій визначається правдоподібність;  $I_k$  – середня амплітуда, яку видає гіпотетичний об'єкт;  $k$  – номер ітерації.

Таким чином, отримане поле  $z$ , яке характеризується значеннями:

$$z_{ij} = \begin{cases} I(t)h_{ij}(x) + \varepsilon, & \text{наявний об'єкт;} \\ \varepsilon, & \text{відсутній об'єкт,} \end{cases}$$

де  $\varepsilon$  – нормально розподілена випадкова величина, що відповідає шуму.

Для вирішення поставленої задачі будується математична модель із застосуванням *методу інтеграції Монте-Карло*.

Вводиться множина маркерів –  $N$  незалежних розподілених випадкових точок  $\{x_{0:t}^{(i)}; i = 1, \dots, N\}$  з відповідною ймовірністю  $p(x_{0:t}|y_{1:t})$ . Тоді емпірична оцінка даного розподілу буде такою:

$$P_N(dx_{0:t}|y_{0:t}) = \frac{1}{N} \sum_{i=1}^N \delta_{x_{0:t}^{(i)}}(dx_{0:t}), \quad (10)$$

де  $\delta_{x_{0:t}^{(i)}}(dx_{0:t})$  позначає дельта-функцію Дірака в точці  $x_{0:t}^{(i)}$ .

Використовуючи (10), отримуємо оцінку для  $I(f_t)$ :

$$I_N(f_t) = \int f_t(x_{0:t}) P_N(dx_{0:t}|y_{1:t}) = \frac{1}{N} \sum_{i=1}^N f_t(x_{0:t}^{(i)}). \quad (11)$$

Якщо апостеріорна дисперсія  $f_t(x_{0:t})$  задовольняє умові  $\sigma_{f_t}^2 \triangleq E_{p(x_{0:t}|y_{1:t})}[f_t^2(x_{0:t})] - I^2(f_t) < +\infty$ , то дисперсія  $I_N(f_t)$  буде дорівнювати  $D(I_N(f_t)) = \frac{\sigma_{f_t}^2}{N}$ . З цього, за законом великих чисел, випливає:

$$I_N(f_t) \xrightarrow[N \rightarrow +\infty]{a.s.} I(f_t). \quad (12)$$

**Побудова фільтру Bootstrap.** Ключова ідея функціонування фільтру *Bootstrap* полягає в усуненні маркерів, що мають низьке значення  $N_t^{(i)}$ , і множенні маркерів з  $N_t^{(i)}$ , що мають більше важливе значення. Для цього емпірична оцінка розподілу визначається, як:

$$P_N(dx_{0:t}|y_{1:t}) = \frac{1}{N} \sum_{i=1}^N N_t^{(i)} \delta_{x_{0:t}^{(i)}}(dx_{0:t}), \quad (13)$$

де  $N_t^{(i)}$  є кількістю утворених маркерів, асоційованих з маркером  $x_{0,t}^{(i)}$  таке, що  $\sum_{i=1}^N N_t^{(i)} = N$ . Якщо  $N_t^{(j)} = 0$ , то маркер  $x_{0,t}^{(j)}$  знищується;  $N_t^{(i)}$  підбираються таким чином, щоб наблизити  $P_N(dx_{0,t}|y_{0,t})$  до  $\hat{P}_N(dx_{0,t}|y_{0,t})$ , а саме: для будь-якої функції  $f_t$  виконується:

$$\int f_t(x_{0,t}) P_N(dx_{0,t}|y_{1,t}) \approx \int f_t(x_{0,t}) \hat{P}_N(dx_{0,t}|y_{1,t}). \tag{14}$$

Після кроку вибору для маркерів  $x_{0,t}^{(i)}$ , що мають  $N_t^{(i)} > 0$ , вважають розподіл відповідно до  $p(x_{0,t}|y_{1,t})$ . Існує багато способів вибору  $N_t^{(i)}$ . В даному алгоритмі вибір маркерів, що переходять до наступної ітерації, проходить за допомогою генерації нової множини розміру  $N$  на основі розподілу  $\hat{P}_N(dx_{0,t}|y_{1,t})$ . Це еквівалентно генерації вибірки з числа нащадків  $N_t^{(i)}$  згідно з багаточленним розподілом параметрів  $\tilde{w}_t^{(i)}$ . Припускається, що (14) виконується, оскільки для будь-якої обмеженої функції  $f_t$ ,  $f_t = \sup_{x_{0,t}} |f_t(x_{0,t})|$ , існує  $C$  така, що

$$E \left[ \left( \int f_t(x_{0,t}) P_N(dx_{0,t}|y_{1,t}) - \int f_t(x_{0,t}) \hat{P}_N(dx_{0,t}|y_{1,t}) \right)^2 \right] \leq \frac{C f_t^2}{N}. \tag{15}$$

Остаточно, фільтр Bootstrap буде мати такий алгоритм:

1) Ініціалізація,  $t=0$ :

- Для  $i=1, \dots, N$ , генерується вибірка  $x_0^{(i)} \sim p(x_0)$  та  $t := 1$ .

2) Вибірка:

- Для  $i=1, \dots, N$ , генерується  $\tilde{x}_t^{(i)} \sim p(x_t|x_{t-1}^{(i)})$  і присвоюється:

$$\tilde{x}_{0,t}^{(i)} = (x_{0,t-1}^{(i)}; i = 1, \dots, N). \tag{16}$$

- Для  $i=1, \dots, N$  оцінюється значення ваги:

$$\tilde{w}_t^{(i)} = p(y_t|\tilde{x}_t^{(i)}). \tag{17}$$

- Виконується нормалізація значення ваг.

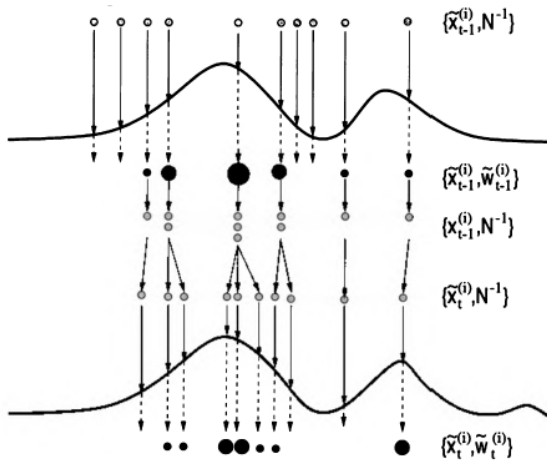


Рис. 4 – Ілюстрація генерації нових маркерів при переході на новий крок.

3) Селекція:

- Генерується заміна  $N$  маркерів  $(x_{0,t}^{(i)}; i = 1, \dots, N)$  з множини  $(\tilde{x}_{0,t}^{(i)}; i = 1, \dots, N)$  з врахуванням ваги.
- Встановлюється  $t \leftarrow t+1$  і виконується перехід до кроку 2.

Зауважимо, що у рівнянні (17) відсутня залежність від  $\tilde{w}_{t-1}^{(i)}$ , оскільки поширені частки  $x_{0,t-1}^{(i)}$  мають однорідні ваги після етапу редискретизації в час  $t-1$ . Крім того, для алгоритму не потребується зберігання траєкторії шляху частинок від часу 0 до часу  $t$ , якщо має значення тільки оцінка  $p(x_t|y_{1,t})$ . Графічне представлення схеми алгоритму показано на рис. 4.

У цьому прикладі фільтр Bootstrap починається в момент часу  $t-1$  з незваженою оцінкою  $\{ \tilde{x}_{t-1}^{(i)}, N^{-1} \}$ , що

забезпечує наближення  $p(x_{t-1}|y_{1,t-2})$ . Для кожного маркеру обчислюється значення ваг, використовуючи інформацію в момент часу  $t-1$ . Це призводить до зваженої оцінки  $\{ \tilde{x}_{t-1}^{(i)}, \tilde{w}_{t-1}^{(i)} \}$ , що дає наближення для  $p(x_{t-1}|y_{1,t-1})$ . На кроці редискретизації вибираються тільки маркери, що найбільш придатні для отримання незваженої оцінки  $\{ x_{t-1}^{(i)}, N^{-1} \}$ , яка все ще є наближенням  $p(x_{t-1}|y_{1,t-1})$ . На кроці дискретизації (прогнозування) вводиться різноманітність, що призводить до  $\{ \tilde{x}_t^{(i)}, N^{-1} \}$ , що є наближенням  $p(x_t|y_{1,t-1})$ .

**Алгоритм виявлення шуканого об'єкту.** Алгоритм, що описується нижче, створює модель поведінки маркерів, які можуть знаходитись в двох станах  $\{1, 2\}$ . Перший стан відповідає маркерам, які рівномірно розподілені по сектору. Другий стан відповідає маркерам що визначають наявність об'єкту. Чим більше маркерів з ознаками 2-го стану будуть зосереджуватись в околі центра мас, тим більше буде імовірність, що в центрі мас клітинки з цими маркерами знаходиться шуканий об'єкт.

1.  $k := 0$ , генерується  $N_0$  маркерів  $\left\{x_0^{(2,l)}\right\}_{l=1}^{N_0}$  з рівномірним розподілом їх по зоні спостереження (зоні пошуку). Ці маркери мають 2-ий стан.
2. Генерується  $M - N_{k-1}$  маркерів  $\left\{x_k^{(1,l)}\right\}_{l=1}^{M-N_{k-1}}$  з рівномірним (чи нормальним) розподілом їх по зоні спостереження (зоні пошуку). Ці маркери мають 1-й стан.
3. З вірогідністю в 0.9 маркери залишаються в тому ж стані, в іншому випадку, вони його змінюють. В результаті чого отримуємо нову множину, яка складається з  $N_k$  маркерів. Перепозначимо її  $\left\{\bar{x}_k^{(2,l)}\right\}_{l=1}^{M-N_k}$ .
4. Знаходяться нормалізовані ваги для кожного маркеру, що має 2-й стан:

$$\mathcal{L}\left(Z_k; I_k, x_k^{(2,l)}, \lambda\right) - \text{функція правдоподібності};$$

$$\mathcal{L}\left(y_k; I_k, \bar{x}_k^{(2,l)}, \lambda\right) = \prod_{ij} e^{-\frac{I_k h_{ij}\left(\bar{x}_k^{(2,l)}\right)\left(I_k h_{ij}\left(\bar{x}_k^{(2,l)}\right) - 2z_{ij}^k\right)}{2\sigma^2}}; \quad (18)$$

$$w^{(i)} = \frac{\mathcal{L}\left(Z_k; I_k, \bar{x}_k^{(2,l)}, \lambda\right)}{\sum_{j=1}^N \mathcal{L}\left(Z_k; I_k, \bar{x}_k^{(2,l)}, \lambda\right)}, \quad l = 1, \dots, N_k. \quad (19)$$

5. Знаходиться  $\hat{x}_k = \sum_{l=1}^{N_k} w_k^{(l)} \bar{x}_k^{(2,l)}$  – положення об'єкту з урахуванням положення центру мас маркерів

$P = \sum_l D\left(x_k^{(2,l)}, \hat{x}_k\right)$  – вірогідність, що в центрі мас знаходиться деякий об'єкт, де  $D(x, y)$  – функція Дірака з центром в  $y$ .

6. Генерується нова множина точок  $\left\{x_k^{(l*)}\right\}_{l=1}^{N_k}$ , де кожний елемент вибирається з множини  $\left\{x_k^{(2,l)}\right\}_{l=1}^{N_k}$  з вірогідністю  $\Pr\left(x_k^{(l*)} = x_k^{(2,j)}\right) = w_k^{(j)}$ .
7. Генерується множина  $\left\{v_k^{(i)}\right\}_{i=1}^N$ , за допомогою нормального розподілу  $N(0, Q)$ , а також знаходиться множина маркерів на наступному кроці:

$$x_{k+1}^{(2,i)} = x_k^{(i*)} + u_k \Delta t + v_k^{(i)} \Delta t, \quad i = 1, \dots, N, \quad (20)$$

де  $Q$  – дисперсія швидкості (за значення можна взяти максимальну швидкість, з якою може рухатись об'єкт);  $u_k$  – швидкість об'єкту на поточному кроці;  $\Delta t$  – час, який пройшов між ітераціями ( $\Delta t = t_k - t_{k-1}$ ).

Швидкість можна знайти за такими формулами:

$$u_t \approx \frac{3\hat{x}_k - 4\hat{x}_{k-1} + \hat{x}_{k-2}}{2\Delta t}, \quad (21)$$

$$u_k \approx \frac{\hat{x}_k - \hat{x}_{k-1}}{\Delta t}, \quad (22)$$

$k := k + 1$  та перейти до кроку 2.

**Адаптація алгоритму для практичного використання.** Алгоритм, що був описаний вище, потребує використання великої кількості обрахунків, що впливає на швидкість виконання програми. Це виникає внаслідок того, що:

- 1) для знаходження  $z_{ij}$  потрібно враховувати всю множину вхідних даних;
- 2) для кращого розпізнавання потрібно запускати алгоритм для великої кількості сегментів одночасно.

Все це за собою тягне генерацію великої кількості маркерів, для якої потрібно знаходити ваги  $w^{(i)}$ .

Для прискорення функціонування алгоритму (з покращенням асимптотичної оцінки швидкості), доцільно реалізувати наступні кроки:

- 1) додати багатопоточність. На першому етапі рахувати величини одночасно по декільком сегментам. На другому – робити одночасно обрахунки для різних множин, оскільки вони не залежні один від одного;
- 2) для  $z_{ij}$ , враховувати сигнали лише в околі, а не по всіх вхідних даних, оскільки вплив сигналу швидко прямує до 0 внаслідок використання функції  $h_{ij}$ .

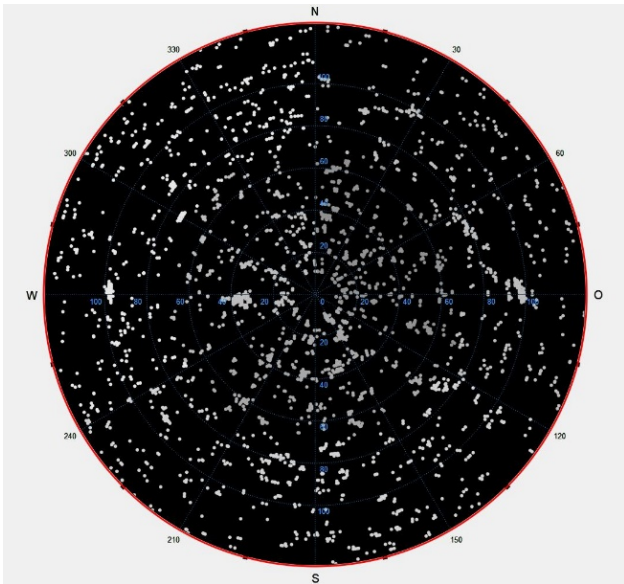


Рис. 5 – Множина маркерів на  $k = 2$  ітерації.

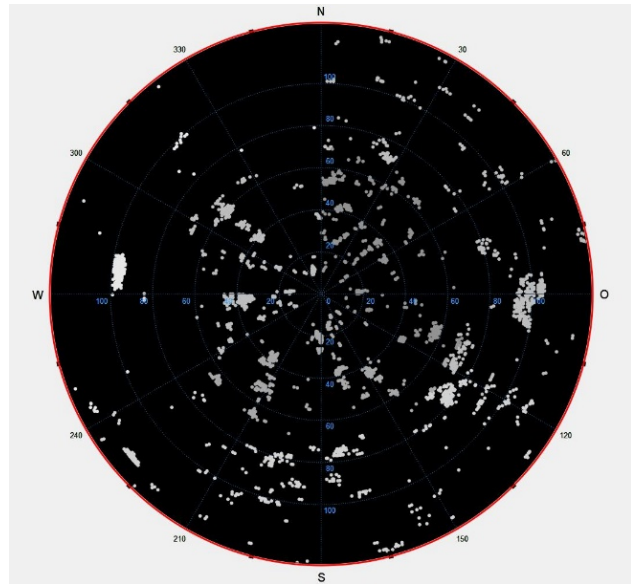


Рис. 6 – Множина маркерів на  $k = 13$  ітерації.

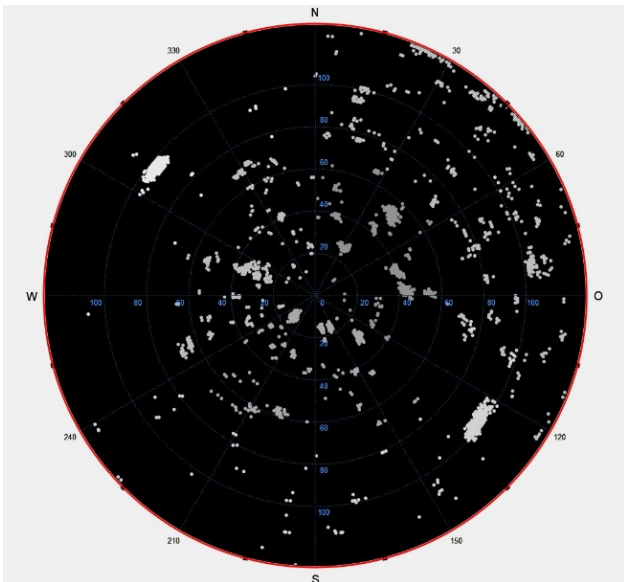


Рис. 7 – Множина маркерів на  $k = 57$  ітерації.

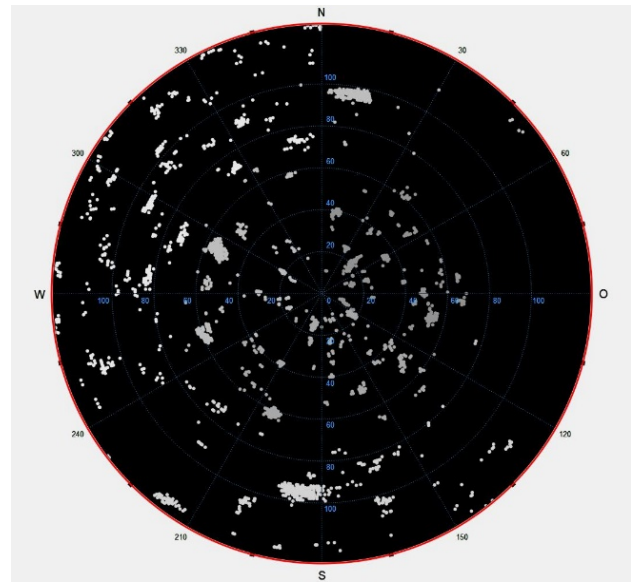


Рис. 8 – Множина маркерів на  $k = 142$  ітерації.

**Результати тестування алгоритму. 1.** Генеруємо 200 шумових сигналів з амплітудою сигналу, яка має рівномірний розподіл та коливається від 200 до 1200. Серед них є сигнали, що відповідають одразу двом об'єктам. Сигнал першого об'єкту має початкове положення – 96 м, 90°. Його амплітуда має рівномірний розподіл та коливається з 500 до 1500. Сигнал другого об'єкту має початкове положення – 96 м, 270°. Його амплітуда має рі-



вномірний розподіл та коливається з 1500 до 2500. Об'єкти рухаються за годинниковою стрілкою, зберігаючи при цьому відстань від центру.

Множини маркерів позначені білим. Траса показується білим слідом, де окремими точками позначені вхідні дані (сигнали) рис. 5 – 8.

2. Генеруємо 1000 шумових сигналів з амплітудою сигналу, яка має рівномірний розподіл та коливається з 200 до 1200. Серед них є сигнали, що відповідають двом об'єктам. Першому об'єкту відповідає 3 сигнали, які мають початкове положення в невеликому околі ( $\pm 5^\circ, \pm 5\text{ м}$ ) точки з положенням – 96 м,  $90^\circ$ . Амплітуди їх мають рівномірний розподіл та коливаються з 500 до 1500. Сигнал другого об'єкту має початкове положення – 96 м,  $270^\circ$ . Його амплітуда має рівномірний розподіл та коливається з 1500 до 2500. Об'єкти рухаються за годинниковою стрілкою, зберігаючи відстань до центру.

Множини маркерів позначені білим. Траса показується білим слідом, де окремими точками позначені вхідні дані (сигнали) рис. 9 – 12.

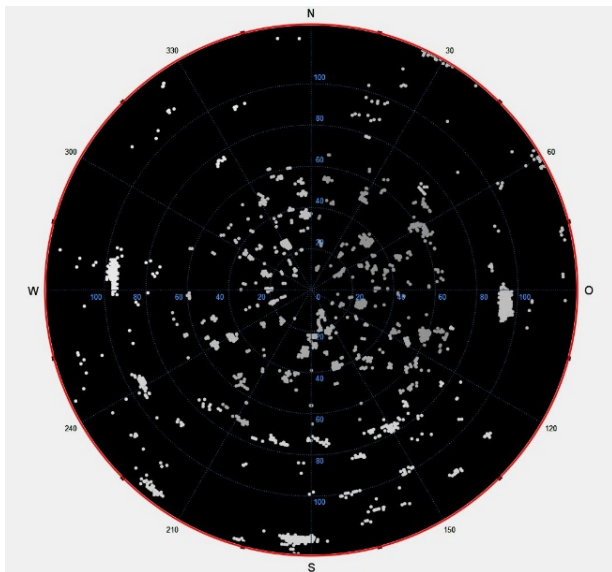


Рис. 9 – Множина маркерів на  $k = 10$  ітерації.

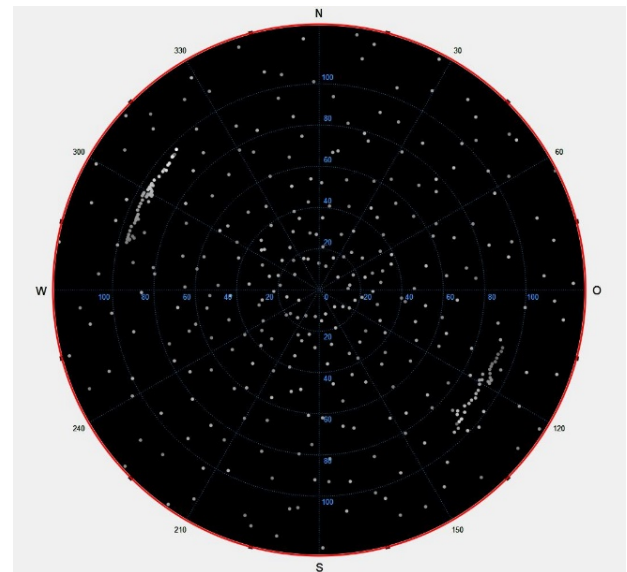


Рис. 10 – Траса об'єкту на  $k = 62$  ітерації.

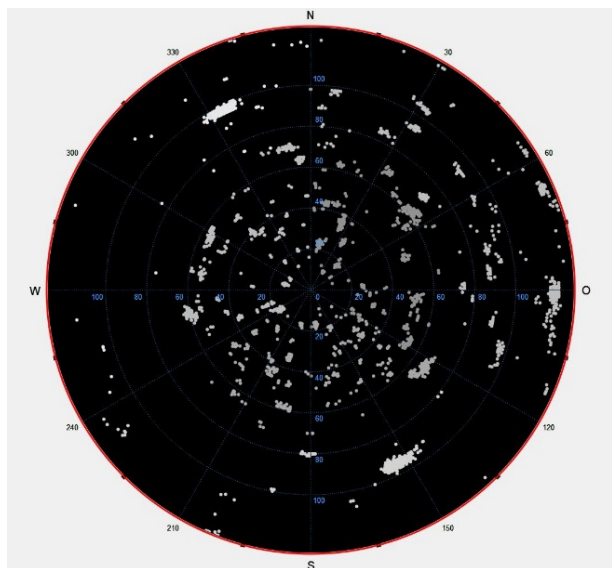


Рис. 11 – Множина маркерів на  $k = 91$  ітерації.

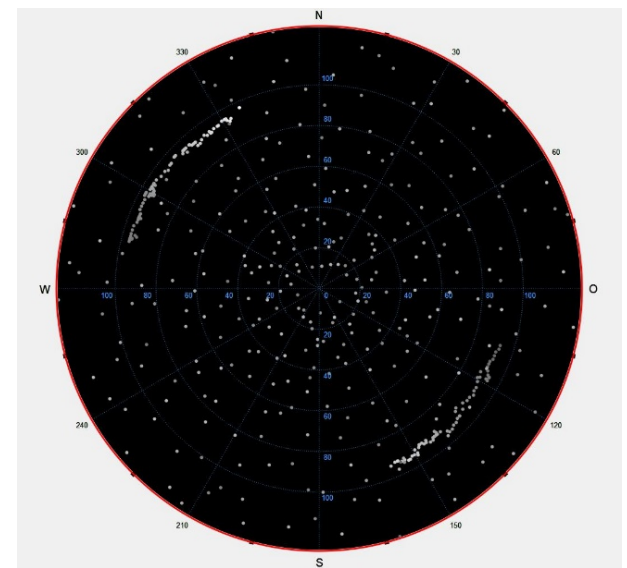


Рис. 12 – Траса об'єкту на  $k = 92$  ітерації.

**Висновки.** Використовуючи метод перетворення вхідних даних від відзеркаленого акустичного сигналу, метод Монте-Карло та технологію фільтра Bootstrap, побудовано технологію розпізнання перешкод. За допомогою штучної моделі, проведено тестування алгоритму. Показано, що алгоритм надає можливість трасувати об'єкт із сильною амплітудою сигналу не втрачаючи його. Показано, що для виявлення об'єкту, який має від-

*Вісник Національного технічного університету «ХПІ». Серія: Математичне моделювання в техніці та технологіях, № 1 ' 2023.*



дзеркалений акустичний сигнал з порівнянням до амплітуди сигналу шуму. Запропонований метод здатен надавати імовірність виявлення перешкоди, близької до ідеальної моделі. Безумовно, алгоритм розпізнання об'єкту по отриманим кластерам маркерів потребує подальшого вдосконалення. Властивістю алгоритму є підвищення імовірності розпізнавання зі збільшенням кількості ітерацій. Алгоритм потребує збільшення кількості імпульсних акустичних сигналів в серії. За законом великих чисел функція розподілу перетворених вхідних даних, а саме – величини  $z_{ij}$ , наближається до нормального, що показано графічно.

Недоліки методу можливо подолати при ускладненні математичної моделі, яка здатна врахувати те, що маркери не завжди зосереджені в центрі мас. Прикладом є об'єкти, з амплітудою сигналу близькою до шуму, де виникнення сигналу має випадковий характер в околі від його дійсного положення. Отриманий кластер маркерів зосереджений у більшому околі, ніж об'єкти з постійним положенням. В результаті перешкода може розпізнаватись саме при використанні функції Дірака, як об'єкт, що погіршує трасування його зміщення.

Для подальшого підвищення ефективності алгоритму потребується вдосконалення математичної моделі для знаходження більш, ніж однієї цілі. Обмеження області обрахунку для величини  $z_{ij}$  та  $w_{ij}$  покращило швидкість алгоритму. Подальше покращення швидкодії досягається при використанні багатопотоковості, де окремий потік буде відповідати окремому сегменту та застосуванні технології розпаралелення для прискорення обчислень.

#### Список літератури

1. Boers Y., Driessen J. N., Vershure F., Heemals W. P. M. H., Juloski A. A Multi Target Track Before Detect Application // Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'03), 16 – 22 June 2003. – Madison, Wisconsin, USA, 2003. DOI: 10.1109/CVPRW.2003.10100.
2. Lu Jia, Ming Li, Lu Xing, Yan Wu, Wanying Song. An improved particle filter for DIM radar target detection and tracking // Published in: IET International Radar Conference, 14 – 16 April 2013. – Xi'an, China, 2013. DOI: 10.1049/cp.2013.0474.
3. Belokurov V. A., Kozlov D. N. The Use of Particle Filtering for Detecting a High Maneuvering Target // 12<sup>th</sup> International Conference on Actual Problems of Electronics Instrument Engineering (APEIE), 2 – 4 Oct 2014. – Novosibirsk, Russia, 2014. DOI: 10.1109/APEIE.2014.7040910.
4. Samuel J. Davey, Mark G. Rutten, Brian Cheung. A Comparison of Detection Performance for Several Track-Before-Detect Algorithm // Journal on Advances in Signal Processing Volume 2008. Published in: 11<sup>th</sup> International Conference on Information Fusion. – Cologne, Germany : Hindawi Publishing Corporation EURASIP, 30 June – 3 July 2008. – Article ID 428036. – 10 pages. DOI: 10.1155/2008/428036.
5. Amaud Doucet, Nando de Freitas, Neil Gordon (ed.). Sequential Monte Carlo methods in practice. – New York : Publisher: «Springer-Verlag», 2001. – 582 p. DOI: 10.1007/978-1-4757-3437-9.

#### References (transliterated)

1. Boers Y., Driessen J. N., Vershure F., Heemals W. P. M. H., Juloski A. A Multi Target Track Before Detect Application. *Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'03)*, 16 – 22 June 2003. Madison, Wisconsin, USA, 2003. DOI: 10.1109/CVPRW.2003.10100.
2. Lu Jia, Ming Li, Lu Xing, Yan Wu, Wanying Song. An improved particle filter for DIM radar target detection and tracking. *Published in: IET International Radar Conference, 14 – 16 April 2013.* – Xi'an, China, 2013. DOI: 10.1049/cp.2013.0474.
3. Belokurov V. A., Kozlov D. N. The Use of Particle Filtering for Detecting a High Maneuvering Target. *12<sup>th</sup> International Conference on Actual Problems of Electronics Instrument Engineering (APEIE)*, 2 – 4 Oct 2014. Novosibirsk, Russia, 2014. DOI: 10.1109/APEIE.2014.7040910.
4. Samuel J. Davey, Mark G. Rutten, Brian Cheung. A Comparison of Detection Performance for Several Track-Before-Detect Algorithm. *Journal on Advances in Signal Processing Volume 2008. Published in: 11<sup>th</sup> International Conference on Information Fusion.* Cologne, Germany, Hindawi Publishing Corporation EURASIP, 30 June – 3 July 2008, Article ID 428036, 10 pages. DOI: 10.1155/2008/428036.
5. *Sequential Monte Carlo methods in practice.* Ed.: Amaud Doucet, Nando de Freitas, Neil Gordon. New York, Publisher: «Springer-Verlag», 2001. 582 p. DOI: 10.1007/978-1-4757-3437-9.

Надійшла (received) 07.04.2023

Відомості про авторів / Сведения об авторах / Information about authors

**Васін Павло Олексійович** – аспірант факультету комп'ютерних наук та кібернетики, Київський національний університет імені Тараса Шевченка, м. Київ; тел.: (099) 765-90-29; e-mail: vasinpavlo@gmail.com.

**Васин Павел Алексеевич** – аспірант факультета компьютерных наук и кибернетики, Киевский национальный университет имени Тараса Шевченко, г. Киев; тел.: (099) 765-90-29; e-mail: vasinpavlo@gmail.com.

**Vasin Pavlo Oleksiiovych** – aspirant of the Faculty of Computer Science and Cybernetics of Taras Shevchenko National University of Kyiv, Kyiv; tel.: (099) 765-90-29; e-mail: vasinpavlo@gmail.com.