НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ
КАФЕДРА ЕЛЕКТРОННИХ ПРИСТРОЇВ ТА СИСТЕМ

«На правах рукопису»
УДК 612.314

«До захисту допущено»
В.о. завідувач кафедри

__________ Юлія ЯМНЕНКО
(підпис) (ініціали, прізвище)

"____"____________2020р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальністю 171 Електроніка
(код і назва)

освітня програма (спеціалізація) Електронні компоненти і системи

на тему: генератор Вінерівського процесу з візуалізацією

Виконав студент II курсу, групи ДС-81мн
(шифр групи)

Мартинюк Вадим Ігорович ____________
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник д.т.н., проф. Жуйков В.Я. ____________
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант ____________ к.т.н., доц., Клен К.С. ____________
(назва розділу) (науковий ступінь, вчене звання, , прізвище, ініціали) (підпис)

Рецензент д.т.н., проф. Найда С.А. ____________
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Консультант
по нормоконтролю к.т.н., доц. Батрак Л.М. ____________
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент ________________
(підпис)

Київ – 2020 року
Національний технічний університет України
“Київський політехнічний інститут
імені Ігоря Сікорського”

Факультет електроніки
(повна назва)

Кафедра Електронних пристроїв та систем
(повна назва)

Рівень вищої освіт – другий (магістерський) за освітньо-науковою програмою

Спеціальність 171 Електроніка
(шифр і назва)

Освітня програма (спеціалізація) Електронні компоненти і системи

ЗАТВЕРДЖУЮ
В.о. завідувач кафедри

___________ Юлія ЯМНЕНКО
(підпис) (прізвище ініціали)

«____» _______2020 року

З А В Д А Н Н Я

НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ СТУДЕНТУ

Мартинюк Вадим Ігорович
(прізвище, ім’я, по батькові)

1. Тема дисертації генератор Вінерівського процесу з візуалізацією
науковий керівник дисертації _________Жуйков В.І. д.т.н., проф._________,
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від «___» лютого 2020 року № 4114-с

2. Термін подання студентом дисертації ________________________________

3. Об’єкт дослідження Математична модель випадкового блукання для моделювання випадкової складової відновлюваних джерел енергії

4. Вихідні дані математична модель генератора випадкового блукання з заданим законом розподілу значень кроку.

5. Перелік завдань, які потрібно розробити 1. Огляд існуючих моделей випадкового блукання. 2.Розробка математичної моделі випадкового блукання з використання з використанням лінійних функцій на базі функцій Уолша. 3. Програмна реалізація розробленої математичної моделі. 4. Порівняння ефективності моделей. 5. Розробка стартап-проекту.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу Презентація

7. Орієнтовний перелік публікацій 1. Prediction of the wind speed change function by linear regression method, Computational problems of electrical engineering. – 2019. Vol. 9, No. 2 С. 28-33 V. Martynyuk, M, Yaremenko. 2. Мартинюк В.І. Лінійні функції на базі функцій Уолша / В.І. Мартинюк, К. С. Клен // Мікросистеми, Електроніка та Акустика. – 2019. – Том 24, №1. – С. 24-39. 3. Мартинюк В.І., Клен К.С., Жуйков В.Я. Моделювання хмарного покриву на основі супутникових знімків, Праці Інституту електродинаміки НАН України. 4. Valery Zhuikov, Vadym Martynyuk, Kateryna Osypenko The linear approximation of wind speed change function IV scientific conference of students "Generation – Transmission – Use GPW 2017", Proceedings. – 2017 C. 31-37 (7 стор.) 5. Мартинюк В.І. «Покращення системи функцій Уолша». Збірник праць XI міжнародної науково-технічної конференції молодих вчених «Електроніка-2018», с. 284-288.

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
технічний	к.т.н., доц., Клен К.С.		

9. Дата видачі завдання _____жовтень 2019 р.________

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Огляд існуючих математичних і програмних моделей випадкового блукання.	10.10.19-15.12.19	
2	Огляд існуючих ортонормованих рядів	16.12.19-23.01.20	
3	Створення математичної моделі випадкового блукання з використанням ортогональних рядів	24.01.20-15.03.20	
4	Програмна реалізація моделі випадкового блукання з використанням мови програмування Python	15.03.20-16.05.20	

Студент _____________ ______________________
(підпис) (ініціали, прізвище)

Науковий керівник дисертації _____________ ______________________
(підпис) (ініціали, прізвище)

АНОТАЦІЯ

Метою даної роботи є розробка алгоритму роботи та комп'ютерна реалізація генератора випадкового блукання на основі лінійних функцій на базі функцій Уолша.

Для збільшення енергоефективності систем розосередженої генерації необхідно проводити аналіз і моделювання на етапі проектування для врахування максимальної кількості параметрів. Одним з найважливіших параметрів, які суттєво впливають на ефективність роботи, є мінливість джерел відновлюваної енергії. Для можливості врахування подібної складової доцільно використовувати процеси, які містять подібний стохастичний характер. Для цього в даній роботі використовується математична модель випадкового блукання з використанням ортогональних рядів. В якості подібних рядів були обрані лінійні функції на базі функцій Уолша, які були отримані за допомогою використання процедури Грама-Шмідта до звичайної системи функцій Уолша.

В результаті дослідження було розроблено математичну модель випадкового блукання з використанням лінійних функцій на базі функцій Уолша, а також програмну реалізацію даної моделі за допомогою мови програмування Python.

Ключові слова: система розосередженої генерації, відновлювані джерела енергії, випадкове блукання, функції Уолша, процедура Грама-Шмідта.

АННОТАЦИЯ

Целью данной работы является разработка алгоритма работы и компьютерная реализация генератора случайного блуждания на основе линейных функций на базе функций Уолша.

Для увеличения энергоэффективности систем рассредоточенной генерации необходимо проводить анализ и моделирование на этапе проектирования для учета максимального количества параметров. Одним из важнейших параметров, которые существенно влияют на эффективность работы, является изменчивость источников возобновляемой энергии. Для возможности учета подобной составляющей целесообразно использовать процессы, которые содержат подобный стохастический характер. Для этого в данной работе используется математическая модель случайного блуждания с использованием ортогональных рядов. В качестве подобных рядов были выбраны линейные функции на базе функций Уолша, которые были получены с помощью использования процедуры Грама-Шмидта к обычной системе функций Уолша.

В результате исследования была разработана математическая модель случайного блуждания с использованием линейных функций на базе функций Уолша, а также программную реализацию данной модели с помощью языка программирования Python.

Ключевые слова: система рассредоточенной генерации, возобновляемые источники энергии, случайное блуждание, функции Уолша, процедура Грама-Шмидта.

SUMMARY

The aim of this work is to develop an operation algorithm and a computer implementation of a random walk generator based on linear functions based on Walsh functions.

To increase the energy efficiency of dispersed generation systems, it is necessary to perform analysis and simulation at the design stage to take into account the maximum number of parameters. One of the most important parameters that significantly affect work efficiency is the variability of renewable energy sources. For the possibility of accounting for such a component, it is

advisable to use processes that contain a similar stochastic character. To this end, in this work, we use a mathematical model of random walk using orthogonal series. As similar series, linear functions based on Walsh functions were chosen, which were obtained using the Gram-Schmidt procedure to the usual system of Walsh functions.

As a result of the study, a mathematical model of random walk using linear functions based on Walsh functions was developed, as well as a software implementation of this model using the Python programming language.

Keywords: dispersed generation system, renewable energy sources, random walk, Walsh functions, Gram-Schmidt procedure.

ЗМІСТ

ВСТУП

Актуальність теми. В останні роки найважливішим трендом в енергетичних комплексах у всьому світі являється перехід від застарілих моделей організації і управління енергетичним комплексом, в якому переважали централізовані виробники, до новітніх моделей розосередженої генерації, які створюють більш конкурентне середовище на енергетичному ринку. Разом з цим робиться значний акцент на використанні відновлюваних і альтернативних джерел енергії, а також на підвищенні енергоефективності існуючої енергетичної інфраструктури.

Проте відновлювані джерела енергії характеризуються значною мінливістю. Тому при розробці та впровадженні нових потужностей необхідно проводити аналіз ефективності роботи для підвищення енергоефективності з урахуванням різноманітних факторів, серед яких випадкова складова первинного джерела енергії.

Створенню моделей випадкового блукання присвячені роботи Simon Raby, Panos Argyrakis, Mark W Keller.

Тому тема дисертаційного дослідження «Генератор випадкового блукання з візуалізацією є актуальною.

Зв'язок роботи з науковими програмами, планами, темами. Дисертація була підготовлена відповідно до науково-дослідного плану кафедри промислової електроніки Національного технічного університету України "Київський політехнічний інститут ім. Ігоря Сікорського.

Метою дослідження є розробка алгоритму роботи та комп'ютерна реалізація генератора випадкового блукання на основі лінійних функцій на базі функцій Уолша.

Для досягнення поставленої мети були вирішені наступні **завдання**:

- Огляд існуючих моделей випадкового блукання;

- Розробка математичної моделі випадкового блукання з використання з використанням лінійних функцій на базі функцій Уолша;
- Програмна реалізація розробленої математичної моделі;
- Порівняння ефективності моделей;
- Розробка стартап проєкту.

Об'єктом дослідження є впровадження математичної моделі випадкового блукання для моделювання випадкової складової відновлюваних джерел енергії.

Предметом дослідження є вибір простої для комп'ютерної реалізації моделі випадкового блукання.

Методи дослідження. При вирішенні поставлених у роботі завдань для розробки математичної моделі випадкового генератора з використанням лінійних функцій на базі функцій Уолша використовується процедура Грама-Шмідта. Комп'ютерна реалізація моделі виконувалася на персональному комп'ютері з використанням мови програмування Python.

Наукова новизна одержаних результатів.

Вперше запропоновано використання випадкового блукання для моделювання випадкової складової джерел відновлюваної енергетики.

Вперше запропоновано використання ортогональних рядів в моделі випадкового блукання. На основі аналізу властивостей даних рядів були обрані лінійні функції на базі функцій Уолша, які мають відносно легку комп'ютерну реалізацію.

Практичне значення одержаних результатів полягає у розробці моделі математичної моделі з використанням лінійних функцій на базі функцій Уолша.

Особистий внесок здобувача. Дисертаційна робота є узагальненням результатів досліджень, проведених автором самостійно. Робота [1] написана автором дисертації особисто. У роботі [2], опублікованій із співавторами, дисертанту належить побудова системи лінійних функцій на базі функцій Уолша за допомогою використання процедури Грама-Шмідта.

Апробація результатів дисертації. Теоретичні положення та результати магістерського дослідження обговорювалися:

1. XI Міжнародна науково-технічна конференція молодих вчених «Електроніка-2018»
2. IV Наукова конференція молодих вчених та студентів «Generation – Transmission – Use GPW-2017»
3. V Наукова конференція молодих вчених та студентів «Generation – Transmission – Use GPW-2018»
4. VI Наукова конференція молодих вчених та студентів «Generation – Transmission – Use GPW-2019»
5. Міжнародна науково-технічна конференція «Проблеми сучасної електроніки – 2020»

Публікації. Основні положення та результати магістерського дослідження висвітлено у наступних публікаціях:

1. Prediction of the wind speed change function by linear regression method, Computational problems of electrical engineering. – 2019. Vol. 9, No. 2 C. 28-33 V. Martynyuk, M, Yaremenko.
2. Мартинюк В.І. Лінійні функції на базі функцій Уолша / В.І. Мартинюк, К. С. Клен // Мікросистеми, Електроніка та Акустика. – 2019. – Том 24, №1. – С. 24-39
3. Мартинюк В.І., Клен К.С., Жуйков В.Я. Моделювання хмарного покриву на основі супутникових знімків, Праці Інституту електродинаміки НАН України
4. Valery Zhuikov, Vadym Martynyuk, Kateryna Osypenko The linear approximation of wind speed change function IV scientific conference of students "Generation – Transmission – Use GPW 2017", Proceedings. – 2017 C. 31-37 (7 стор.)

Структура та обсяг дисертації. Дисертація складається зі вступу, чотирьох розділів, висновків, списку використаних джерел із 51 найменувань

та 1 додатку. Загальний обсяг дисертаційної роботи становить 106 сторінок, у тому числі 87 сторінок основного тексту, 23 рисунки та 20 таблиць.

РОЗДІЛ 1. ОГЛЯД МОДЕЛЕЙ ВИПАДКОВОГО БЛУКАННЯ

Обмежена кількість традиційних паливно-енергетичних ресурсів (газоподібного, рідкого і твердого палива), постійне зростання цін на них, а також негативний вплив продуктів їх згоряння на навколишнє середовище, підштовхують до більш раціонального їх використання, яке може бути досягнуто як за рахунок використання сучасного енергозберігаючого обладнання, так і до створення систем енергозабезпечення споживачів з використанням відновлюваних джерел енергії.

Однак, мала щільність потоку відновлюваної енергії, його мінливість, а також неможливість регулювання режимів її надходження не дозволяють орієнтувати розвиток енергетики виключно на відновлювані джерела енергії. Тому, найбільш економічно ефективними представляються варіанти спільного застосування традиційних енергетичних ресурсів і відновлюваних джерел в складі однієї системи.

Крім того, можна застосувати кілька видів відновлюваних джерел як гібридну систему для задоволення потреб локального споживача. У цьому випадку поєднання таких джерел як вітрове та сонячне забезпечує надійне та економічне вироблення електроенергії у порівнянні з використанням одного джерела.

Змінний і слабо прогнозований характер потужності, властивий вітрової та сонячної енергетики, може привести до негативного впливу на режими роботи енергосистеми. Це, зокрема, стосується стійкості динамічних процесів в системах енергопостачання та відповідно організації управління. Значне впровадження відновлюваних джерел енергії, якщо воно не супроводжується технологіями акумулювання, вимагає додаткового регулювання потужності енергосистеми, щоб збалансувати поточні коливання не тільки споживання, але і енергії, що генерується.

Враховуючи те, що наявні технологічні рішення в галузі відновлюваних джерел енергії майже досягли свого теоретичного максимум, постає питання максимально ефективного використання енергії первинного носія. Перед аналізом відновлюваних джерел енергії в енергосистемах слід врахувати їх стохастичну природу, щоб підвищити точність результатів моделювання при розробці та впровадженні нових потужностей. Тому постає питання створення математичної моделі, яка описуватиме випадкову складову відновлюваних джерел енергії, оскільки необхідно створити належні способи та методи керування для зменшення негативних наслідків випадкової поведінки генераторів на відновлюваних джерелах енергії.

1.1. Випадкові події

Будь-які фізичні явища і процеси мають певні закономірності. Ці закономірності відображаються математичними і фізичними моделями, які підтверджуються або спростовуються проведенням експериментів. Математична модель виражається у вигляді будь-яких формул, співвідношень. Задачею фізичної моделі є відтворення певних властивостей або характеристик реальних об’єктів, явищ, процесів. Будь-яка модель відображає основні і найбільш важливі сторони досліджуваного явища. При цьому існують властивості досліджуваного явища, які описати або дослідити складно або й взагалі не можливо. При експериментальній перевірці моделей можливі два результати, які поділяють всі явища на детерміновані і випадкові [3].

Теорія імовірностей – наука, яка вивчає закономірності випадкових подій. Одним з основних понять теорії ймовірностей є поняття випадкової події. Під випадковою подією в теорії імовірності розуміють будь-яке явище, яке може як відбутися, так і не відбутися при виконанні деякого комплексу умов, що забезпечують незмінність умов проведення експериментів. Поява

кожного такого явища називається дослідом, експериментом або ж випробуванням [4].

Предметом дослідження теорії імовірності є математичні моделі випадкових процесів і явищ, тоді як статистика займається дослідженням реальних процесів і явищ.

Події позначаються великими літерами латинського алфавіту A, B, C; або A_1, A_2, A_3.

1.1.1. Класифікація випадкових подій

Події поділяються на достовірні, неможливі і випадкові.

Достовірною називають подію, яка в результаті проведення експерименту обов'язково відбудеться.

Подія називається неможливою, якщо вона не може відбутися в умовах даного експерименту.

Подія називається можливою, або випадковою, якщо в результаті проведення експерименту вона може як відбутися, так і не відбутися. При цьому повинен мати місце принциповий критерій випадковості, який виключатиме вплив детермінованих факторів. Випадкова подія – наслідок випадкових факторів, вплив яких передбачити неможливо або вкрай важко.

Подібне визначення подій є емпіричним. Надалі будуть надані більш строгі математичні (теоретико-множинні) визначення.

Важливим поняттям є повна група подій. Кілька подій в даному випадковому експерименті утворюють повну групу, якщо в результаті його проведення обов'язково відбудеться хоча б одна з них.

Також розрізняють події сумісні і несумісні. Події називаються сумісними, якщо поява однієї з них не виключає появи іншої. В іншому випадку події називаються несумісними. Найпростішим прикладом несумісних подій є пара протилежних подій. Під протилежною подією розуміється подія, яка обов'язково має відбутися, якщо не настала певна

подія. Подія, протилежна події A, зазвичай позначається рискою вгорі – $\overline{A}$. Протилежні події несумісні і разом вони утворюють повну групу подій.

Інша важлива характеристика подій – це їх рівноможливість. Дві або більше подій називають рівноможливими, якщо поява жодної з них не є більш можливою, ніж інших. В іншому випадку події є нерівноможливими.

Множина несумісних подій утворює повну групу подій, якщо в результаті окремо взятого експерименту обов'язково з'явиться одна з цих подій.

Попарно несумісними подіями називається група подій, кожна з яких є несумісною з усіма іншими подіями даної групи.

Ще одне важливе поняття – це елементарність події. Елементарною називають подію, яка не можна бути факторизована (розкладена) на більш елементарні події.

Простою подією називають подію, яка складається лише з однієї елементарної події. Складна подія складається з двох і більше простих подій.

Наступним кроком буде введення понять, які дозволять строге математичне визначення імовірності, яке знадобиться надалі.

1.1.2. Простір елементарних подій

Перший крок при побудові імовірнісної моделі реального явища або процесу – виділення всіх окремо можливих результатів експерименту (результатів спостережень), які називаються елементарними подіями. Припускається, що експеримент може завершитись лише однією подією.

Простором елементарних подій Ω називається множина, що містить всі можливі результати даного випадкового експерименту, з яких в експерименті відбувається рівно один. Елементи цієї множини являються елементарними подіями і позначаються ω. Простір елементарних подій являється достовірною подією і утворює повну групу подій. При цьому елементарні події є попарно несумісні.

Кажуть, що в результаті експерименту відбулася подія $A \subseteq \Omega$, якщо в експерименті відбулася одна з елементарних подій, що входять в множину A.

Простір елементарних подій буває трьох типів:

1. Скінченним.
2. Зліченним.
3. Незліченним.

Скінченним простором елементарних подій (ПЕП) називають множину, кількість елементів якого скінченна й існує невід'ємне ціле число k (потужність множини), яке рівне кількості елементів даної множини. Тобто існує наступне бієктивне відображення (рис. 1.1, а):

$$\exists f : S \to \{1,...,k\}.$$

Скінченною множиною є, наприклад, множина $\{1,2,3,4,5\}$.

Зліченним ПЕП називають нескінченну множину, елементи якої можна пронумерувати. Тобто існує бієктивне відображення (рис. 1.1, а) всіх елементів зліченної множини на множину натуральних чисел N:

$$\exists f : S \to \{1,...,N\}.$$

Що еквівалентно тому, що потужність множини є рівною потужності множини натуральних чисел. Прикладом зліченної множини є множина натуральних, цілих чисел.

Зліченні і скінченні множини називаються дискретними.

Незліченним ПЕП називають множину, кількість елементів якої настільки велика, що їх не можна пронумерувати. Кажуть, що множина є незліченою, якщо її потужність більша за потужність множини натуральних чисел N. Формальною умовою незліченності для множини є відсутність ін'єктивного відображення (рис. 1.1, б) даної множини на множину натуральних чисел N, а також відсутність сур'єктивного відображення (рис. 1.1, б) множини натуральних чисел N на дану множину:

$$\neg\exists f : S \to \{1,...,k\}.$$

$$\neg\exists f : S \to \{1,...,k\}.$$

Прикладом незліченної множини є множина всіх точок відрізка $[0;1]$.

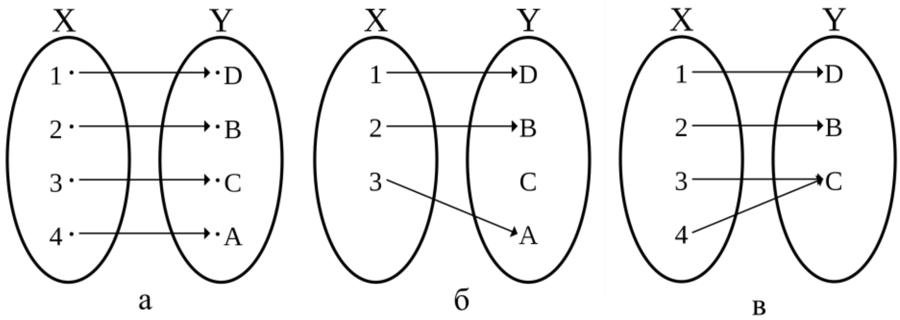


Рис. 1.1. Бієктивне (а), ін’єктивне (б), сур’єктивне відображення (функція)

Розглянемо набір можливих операцій, визначених для простору елементарних подій.

1.1.3. Операції над подіями

Однією з основних задач теорії імовірності являється знаходження імовірності складних подій по відомим імовірностям деяких інших подій. Це стає можливим, якщо будь-які події можна виразити через елементарні події за допомогою різних операцій. Для цього в теорії імовірності використовуються аналоги операцій над множинами.

Добутком (перетином) подій A і B називається так подія C, яка містить елементарні події, які одночасно входять в подію A і в подію B (рис. 1.2, а):

$$C = A \cdot B = \{x_i \mid x_i \in A \wedge x_i \in B\}.$$

Сумою (об’єднанням) подій A і B називається так подія C, яка містить елементарні події, які входять або в подію A або в подію B (рис. 1.2, б):

$$C = A + B = \{x_i \mid x_i \in A \vee x_i \in B\}.$$

Різницею подій A і B називається так подія C, яка містить елементарні події, які входять в подію A і не входять в подію B (рис. 1.2, в):

$$C = A - B = A \setminus B = \{x_i \mid x_i \in A \wedge x_i \notin B\}.$$

Запереченням (протилежною подією) для події A є подія $\overline{A}$, яка відбувається тоді, коли не відбувається A (рис. 1.2, г). Тобто, якщо одночасно добутком даних множин є порожня множина, а сумом – простір елементарних подій Ω (повна група подій):

$$A + \overline{A} = \Omega,$$

$$A \cdot \overline{A} = \varnothing.$$

Дані операції утворюють базис і всі інші операції над подіями можуть бути виражені за допомогою даних операції.

Враховуючи наведені вище операції над подіями, попередню класифікацію випадкових подій можна визначити через операції з множинами і діаграми Венна.

Події A і B є несумісними, якщо їх добуток є порожньою множиною:

$$A \cdot B = \varnothing.$$

Події A і B сумісні, якщо їх добуток не є порожньою множиною. Або, інакше кажучи, потужність добутку подій A і B додатна:

$$|A \cdot B| > 0.$$

Події $A_1 \ldots A_n$ є попарно несумісними, якщо для будь-яких $i \neq j$, де $1 \leq i, j \leq n$:

$$A_i \cdot A_j = \varnothing.$$

Достовірною подією є весь простір елементарних подій Ω, оскільки експеримент завершується хоча би однією елементарною подією. Неможлива подія – це порожня множина (позначається $\varnothing$ або N), оскільки в ньому немає ні одної елементарної події.

Події A і B є еквівалентними (рівнозначними), якщо всі елементарні події, які входять в A, входять і в подію B:

$$A \sim B \Leftrightarrow \{x_i \mid x_i \in A \wedge x_i \in B\}.$$

Рівноможливими називаються події, якщо їх потужності рівні.

Дані співвідношення між множинами відображаються діаграмами Венна (рис. 1.2).

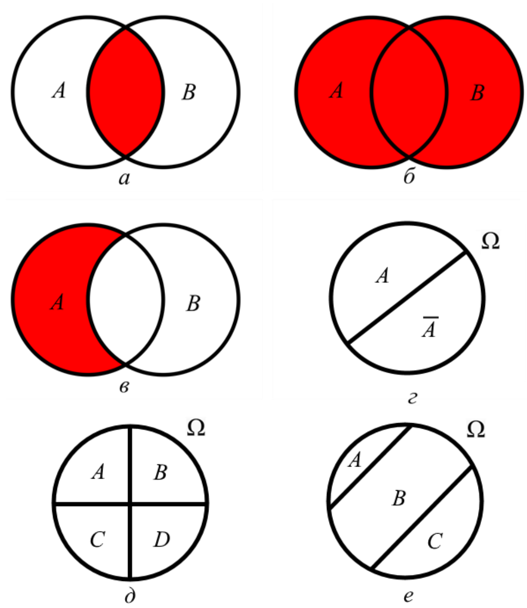


Рис. 1.2. Добуток (а), сума (б), різниця (в) подій A і B; несумісні події (г-е); сумісні події (а-в); рівноможливі події (г, д); нерівноможливі події (е)

Розлянутого теоретичного матеріалу достатньо для математичного визначення терміну імовірності – одного з основних понять теорії імовірностей і випадкових процесів.

1.1.4. Імовірність події. Способи визначення

Імовірність події – чисельна міра ступеню об’єктивної можливості появи даної події. В якості одиниці вимірювання імовірності прийнята імовірність достовірної події. Імовірність неможливої події рівна нулю. Імовірність випадкової події A позначається $P(A)$ і змінюється в межах $0 \le P(A) \le 1$ [4].

В теорії імовірності існує кілька способів формального математичного визначення імовірності випадкової події. Перше з них це статистичне.

Нехай була проведена серія з експериментів n (n називають довжиною серії), в кожному з яких може відбутись або не відбутись подія A. Нехай кількість появи події A в даній серії становить $n(A)$ разів. Тоді відносною частотою події A називається величина:

$$P_n(A) = \frac{n(A)}{N}.$$

При невеликій кількості експериментів відносна частота події має випадковий характер і може суттєво змінюватися від однієї групи дослідів до іншої. При збільшенні кількості експериментів частота $P_n(A)$ стабілізується і приближається до деякого середнього значення. Даний емпіричний факт називається властивістю статистичної стійкості частоти: по мірі необмеженого росту кількості однорідних і незалежних експериментів відносна частота досліджуваної події прямує до деякого постійного значення (рис. 1.3).

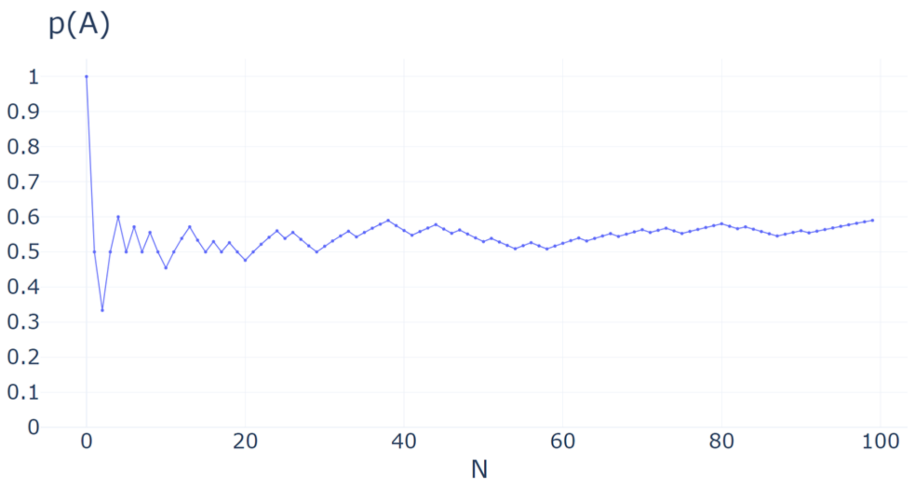


Рис. 1.3. Залежність відносної частоти події A від кількості експериментів N

Число, до якого прямує частота, можна прийняти за імовірність події A, тобто статистична (частотна) інтерпретація імовірності полягає в тому, за приблизне значення імовірності події приймають її відносну частоту.

Математично дана властивість має вид:

$$\lim_{n\to\infty} P_n(A) = P(A).$$

Проте наближення частоти події до його імовірності не являється звичайною збіжністю до границі. Тому для описання збіжності відносної частоти події до його імовірності в теорії імовірностей вводять термін збіжності за імовірністю (за мірою) [4, 5].

Послідовність X_n сходиться за імовірністю до величини a, якщо при будь-якому нескінченно малому $\varepsilon > 0$ імовірність нерівності $|X_n - a| < \varepsilon$ зі збільшенням n прямує до одиниці, тобто $\lim_{n\to\infty} P(|X_n - a| < \varepsilon) = 1$. Позначається як $X_n \xrightarrow{P} a$ [5].

Частота події A відрізняється від імовірності даної події тим, що імовірність є детермінованою величиною, а частота – випадковою величиною. Частотна інтерпретація має ряд недоліків (необхідно проводити велику кількість експериментів для визначення імовірності), тому часто використовують інші визначення імовірності.

Класичною імовірністю події A називається відношення кількісті $n(A)$ елементарних подій, які входять в подію A, до загальної кількості N всіх можливих елементарних подій [5-7]:

$$P(A) = \frac{n}{N}.$$

Що є еквівалентним відношенню потужностей множини A та простору елементарних подій:

$$P(A) = \frac{|A|}{|\Omega|}.$$

При цьому до простору елементарних подій та події Aставляться наступні вимоги:

1. Всі елементарні події простору елементарних подій повинні бути рівноможливими.
2. ПЕП повинен бути скінченним або зліченним, тобто дискретним.

3. Подія A повинна складатися з несумісних елементарних подій.

На практиці визначення імовірності по класичній схемі зводиться до підрахунку загального числа елементарних подій і числа сприятливих подій, що вирішується за допомогою використання елементарних формул комбінаторики.

Класичне визначення імовірності характеризується простотою та наочністю. Проте для визначення імовірності події при нескінченному ПЕП вона не підходить. Для цього використовують геометричну імовірність [4, 5].

Нехай ПЕП визначений в R^m області (лінія, площина, об'єм) з певною скінченною мірою. Тоді імовірність події A визначається наступним чином:

$$P(A) = \frac{\mu(A)}{\mu(\Omega)},$$

де μ – функція, яка визначає міру множини (довжину, площу, об'єм).

Оскільки елементарні події ПЕП, як і у попередньому визначенні, рівноможливі, то імовірність події A не залежить від її форми, а залежить лише від її міри (рис. 1.4).

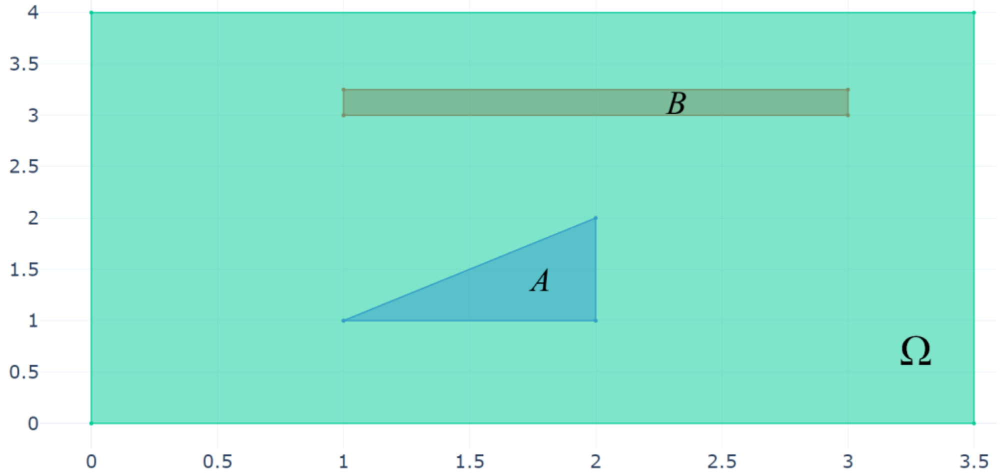


Рис. 1.4. Визначення геометричної імовірності

Класичне визначення імовірності має недолік, оскільки простір елементарних подій складається з рівноможливих подій. Багато моделей не відносяться до дискретних моделей зі зліченною або скінченною кількістю

елементарних подій, з якої складається простір елементарних подій Ω. Геометрична ж інтерпретація не є універсальною, оскільки існують задачі, які не можуть бути вирішені геометрично.

Універсальним визначенням імовірності є аксіоматичне визначення, запропоноване А.М. Колмогоровим.

1.1.5. Алгебра подій. Аксіоматичне визначення імовірності

Нехай задано довільний прості елементарних подій Ω, а θ – деяке сімейство підмножин Ω. Тоді алгеброю подій називається будь-яка непорожня система підмножин простору Ω замкнена відносно зліченного числа теоретико-множинних операцій (наведені в 1.1.3), і якщо виконуються наступні вимоги [7, 8]:

1. $\Omega \in \theta$ – алгебра подій містить достовірну подію.

2. Якщо $A \in \theta$, то протилежна подія $\overline{A} \in \theta$ – разом з будь-якою подією алгебра містить протилежну подію.

3. Якщо події $A_1, ..., A_n \in \theta$ при $n > 2$, то і:

$$\bigcup_{i=1}^{n} A_i \in \theta. \quad (1)$$

Наслідком перших двох умов є те, що алгебра подій обов'язково містить неможливу подію $\varnothing \in \theta$.

Дане визначення алгебри подій є аналогом визначення алгебри множин з теорії множин.

Множина подій називається замкненою відносно операцій, якщо результатом застосування даних операцій до елементів множини є елемент даної множини [9]. Взагалі достатньо, щоб простір елементарних подій був замкнений відносно будь-яких двох бінарних операцій над множинами, із чого відразу слідує те, що дана множина буде замкненою відносно інших операцій над множинами. Проте в аксіоматиці Колмогорова прийнато використовувати різницю та об'єднання [10].

В теорії імовірностей часто виникає потреба проводити об’єднання скінченної кількості подій, вважаючи результат подібного об’єднання теж подією. Розширюючи (1) до об’єднання скінченної кількості подій, отримуємо σ-алгебру подій.

Більш формально, σ-алгеброю подій називають сімейство множин θ, елементами якого є підмножини простору елементарних подій Ω, і якщо виконуються наступні вимоги [10]:

1. $\Omega \in \theta$ – алгебра подій містить достовірну подію.

2. Якщо $A \in \theta$, то протилежна подія $\overline{A} \in \theta$ – разом з будь-якою подією алгебра містить протилежну подію.

3. Якщо події $A_1, A_2 ... \in \theta$, то і $\bigcup\limits_{i=1} A_i \in \theta$.

Тепер використовуючи наведену вище аксіоматику, можна визначити загальне поняття імовірності події.

Імовірністю події (імовірнісною мірою) називається числова функція, яка визначена на σ-алгебрі подій θ і яка кожній події $A \in \theta$ ставить у відповідність значення $P(A)$ так, що виконуються наступні аксіоми [10, 11]:

1. Аксіома невід’ємності – $P(A) > 0$ для всіх $A \in \theta$.

2. Аксіома нормованості – $P(\Omega) = 1$

3. Аксіома скінченної адитивності – для будь-яких попарно несумісних подій в послідовності $A_1, ... A_n$:

$$P\left(\bigcup_{i=1}^{n} A_i\right) = \sum_{i=1}^{n} P(A_i).$$

4. Аксіома зліченної адитивності – для будь-яких попарно несумісних подій в послідовності $A_1, A_2, ...$:

$$P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i).$$

Важливим поняттям в теорії імовірностей з використанням аксіоматики Колмогорова є імовірнісний простір. Імовірнісним простором називається набір:

$$\{\Omega, F, P\},$$

де Ω – простір елементарних подій; F – сигма алгебра подій (підмножин) простору елементарних подій Ω; P – імовірнісна міра подій.

Наступним кроком буде визначення поняття випадкової величини, а також її основних параметрів, які наділі використовуватиметься для створення і описання математичної моделі стохастичної складової відновлюваних джерел енергії.

1.2. Випадкові величини

Нехай задано імовірнісний простір $\{\Omega, F, P\}$. Випадковою величиною ξ називається будь-яка дійсна функція $\xi = \xi(\omega)$, визначена на просторі елементарних подій Ω, і областю значень $R = (-\infty, +\infty)$ [11].

В результаті проведення експерименту випадкова величина може прийняти те чи інше значення, яке заранння невідоме.

Якщо множина можливих значень даної функції скінченна або зліченна, то така випадкова величина називається дискретною. У випадку незліченної множини значень випадкова величина називається неперервною.

Випадкові величини позначаються $\xi_1, \xi_2, \xi_3, ...$ або $\xi, \eta, \varsigma, ...$, а відповідні значення даних величин $x_1, x_2, x_3, ...$ або $x, y, z, ...$.

1.2.1. Функція розподілу

Функцією розподілу випадкової величини ξ називається функція $F_{\xi}(x)$, яка визначена для будь-якого дійсного x і виражає імовірність того, що випадкова величина ξ приймає значення $\xi < x$ [11, 12]:

$$F_{\xi}(x) = P(\xi < x).$$

По неперервності функції розподілу випадкові величини поділяють на:

1. Дискретні
2. Абсолютно неперервні
3. Неперервні (змішані).

Випадкова величина називається дискретною, якщо її функція розподілу кусково-неперервна, а випадкова величина приймає кінцеву або зліченну кількість значень (рис. 1.5, а) [11-13].

Випадкова величина ξ називається абсолютно неперервною, якщо існує невід'ємна функція $p_{\xi}(x)$ така, що при будь-яких значеннях x функцію розподілу можна представити у вигляді інтегралу:

$$F_{\xi}(x) = P(\xi < x) = \int_{-\infty}^{x} p_{\xi}(x)dx,$$

де $p_{\xi}(x)$ – функція густини імовірності або щільності розподілу неперервної випадкової величини.

Функція розподілу немає точок розриву, а випадкова величина приймає незліченну кількість значень і зазвичай являє собою деякий інтервал значень (рис. 1.5, б).

Випадкова величина ξ називається неперервною або змішаною, якщо її функція розподілу $F_{\xi}(x)$ неперервна і має скінченну кількість точок розриву. Множина значень неперервної випадкової величини незліченна і зазвичай являє собою деякий інтервал значень (рис. 1.5, в). Функція розподілу неперервної випадкової величини володіє тими же властивостями, що й дискретна випадкова величина.

Будь-яка функція розподілу володіє наступними властивостями:

1. Для будь-якого $x \in R$ справедлива нерівність $0 \le F_\xi(x) \le 1$.

2. Функція розподілу – неспадна функція, тобто якщо $x_1 \le x_2$, то $F(x_1) \le F(x_2)$.

3. Справедливі наступні граничні умови:

$$\lim_{x \to -\infty} F_\xi(x) = 0,$$
$$\lim_{x \to +\infty} F_\xi(x) = 1.$$

4. Функція розподілу неперервна зліва:

$$\lim_{x \to a-0} F_\xi(x) = F_\xi(a).$$

Функція розподілу являється універсальним законом розподілу випадкової величини.

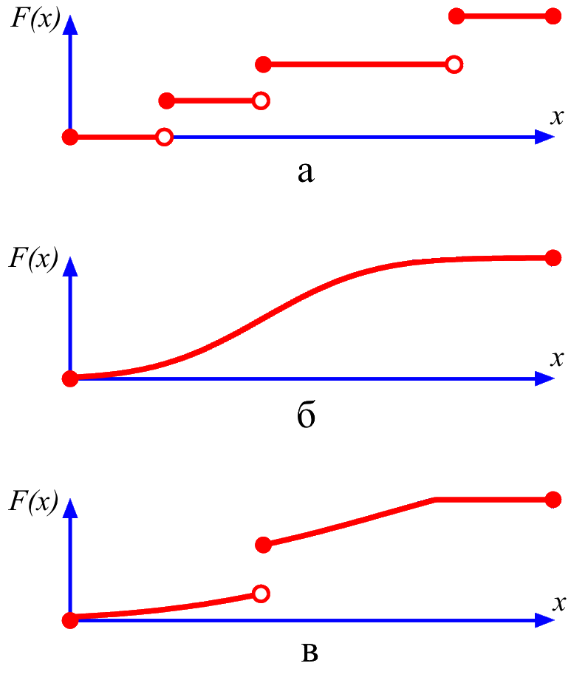


Рис. 1.5. Функція розподілу дискретної (а), абсолютно неперервної (б) і неперервної (в) функції розподілу випадкової величини

Для функції розподілу справедливі наступні співвідношення:

$$P(\xi > x) = 1 - F_\xi(x),$$
$$P(\xi < x) = F_\xi(x - 0),$$
$$P(\xi \geq x) = 1 - F_\xi(x - 0),$$
$$P(\xi = x) = F_\xi(x) - F_\xi(x - 0),$$
$$P(x_1 < \xi \leq x_2) = F_\xi(x_2) - F_\xi(x_1),$$
$$P(x_1 \leq \xi < x_2) = F_\xi(x_2) - F_\xi(x_1 - 0),$$
$$P(x_1 < \xi < x_2) = F_\xi(x_2 - 0) - F_\xi(x_1),$$
$$P(x_1 \leq \xi < x_2) = F_\xi(x_2 - 0) - F_\xi(x_1 - 0).$$

1.2.2. Закон розподілу. Щільність імовірності

Законом розподілу випадкової величини називається будь-яке співвідношення, яке встановлює зв'язок між можливими значеннями випадкової величини і відповідними імовірностями.

Найпростішою формою закону розподілу дискретної випадкової величини являється ряд розподілу (табличний спосіб задання).

Рядом розподілу дискретної випадкової величини ξ називається таблиця, в якій перераховані всі можливі значення $x_1,...x_n$ даної випадкової величини і імовірності, які їм відповідають $p_i = P(\xi = x_i)$ (табл. 1).

Табл. 1.1. Ряд розподілу дискретної випадкової величини

ξ	x_1	x_2	...	x_n
P	p_1	p_2	...	p_n

При цьому сума імовірностей рівна одиниці $\sum_{i=1}^{n} p_i = 1$. Якщо множина значень випадкової величини являється зліченною, то дана таблиця нескінченна справа, а сумою імовірностей являється ряд $\sum_{i=1}^{\infty} p_i = 1$ [13].

Іншим способом задання закону розподілу є графічний спосіб, який ще називають багатокутником розподілу (рис. 1.6).

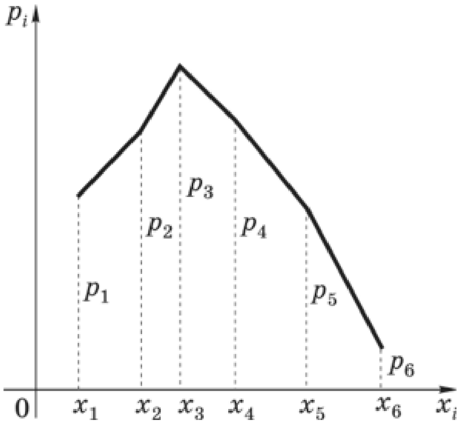


Рис. 1.6. Багатокутник розподілу дискретної випадкової величини

Поставити у відповідність всім значенням неперервної випадкової величини імовірності неможливо, оскільки в такому випадку імовірність появи будь-якого значення випадкової величини буде рівна нулю, що буде показано пізніше. Тому для неперервних випадкових величин вводять поняття функції густини імовірності або щільності неперервної випадкової величини. Позначається $p_\xi(x)$ і визначається:

$$p_\xi(x) = F'_\xi(x).$$

Функція густини імовірності володіє наступними властивостями:

1. Функція додатна – $p_\xi(x) > 0$.

2. В точках неперервності щільність розподілу рівна похідній від функції розподілу $p_\xi(x) = F'_\xi(x)$.

3. Інтеграл по всій числовій осі від функції щільності імовірності дорівнює одиниці:

$$\int_{-\infty}^{+\infty} p_\xi(x) dx = 1.$$

Щільність імовірності розподілу визначає закон розподілу випадкової величини, так як визначає імовірність потрапляння випадкової величини у будь-який напівінтервал:

$$P\left(\xi \in [a,b]\right) = P\left(a \le \xi \le b\right) = \int_a^b p_\xi(x)dx .$$

Щільність імовірності може мати розриви в деяких точках, де її можна визначити. Це ніяк не впливає на функцію розподілу й інші числові характеристики. Графік щільності розподілу називають кривою розподілу (рис. 1.7).

За допомогою щільності імовірності легко показати, що імовірність появи деякого значення неперервної випадкової величини рівна нулю, оскільки верхня і нижня межі інтеграла співпадають:

$$p(\xi = a) = P(\xi = a) = \int_a^a p_\xi(x)dx = 0 .$$

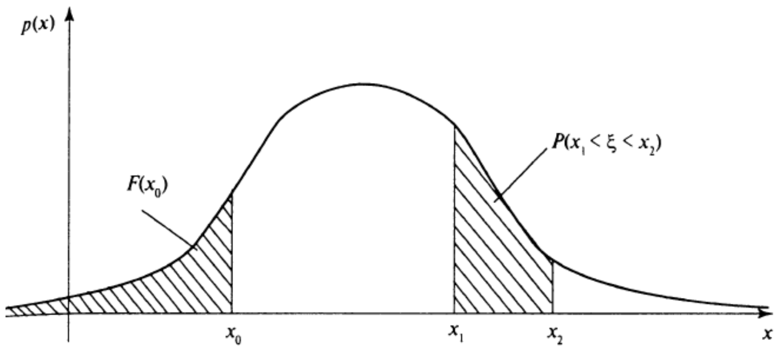


Рис. 1.7. Крива розподілу абсолютно неперервної випадкової величини

1.2.3. Числові характеристики і характеристики залежностей

Для однозначного визначення випадкової величини використовують числові характеристики.

Математичним очікуванням (математичним сподіванням, середнім значенням) дискретної випадкової величини ξ називається число [14]:

$$M(\xi) = \sum_{i=1}^{n} x_i p_i .$$

У випадку зліченої множини значень випадкової величини ξ математичне очікування визначається як нескінченний ряд (якщо він абсолютно збіжний) [14]:

$$M(\xi) = \sum_{i=1}^{\infty} x_i p_i .$$

Математичне очікування для неперервної випадкової величини ξ визначається по формулі [14]:

$$M(\xi) = \int_{-\infty}^{-\infty} x p_{\xi}(x) dx .$$

При цьому інтеграл повинен бути абсолютно збіжним.

Властивості математичного очікування:

1. $M(C) = C, \quad C = const$.
2. $M(C\xi) = CM(\xi)$.
3. $M(\xi_1 + \ldots + \xi_n) = \sum_{i=1}^{n} M(\xi_i)$.
4. $M(\xi_1 \cdot \ldots \cdot \xi_n) = \prod_{i=1}^{n} M(\xi_i)$, якщо $\xi_1 \cdot \ldots \cdot \xi_n$ попарно незалежні.

Дисперсією випадкової величини ξ називається число:

$$D(\xi) = M\left(\xi - M(\xi)^2\right).$$

Величина $\sigma = \sqrt{D(\xi)}$ називається середньоквадратичним відхиленням.

Із визначення дисперсії витікають формули для обчислення дисперсії дискретної випадкової величини зі зліченною кількістю значень [14]:

$$D(\xi) = \sum_{i=1}^{n} \left(x_i - M(\xi)\right)^2 p_i .$$

і для дискретної випадкової величини зі зліченною кількістю значень при умові абсолютної збіжності ряду:

$$D(\xi) = \sum_{i=1}^{\infty} \left(x_i - M(\xi)\right)^2 p_i .$$

Дисперсія неперервної випадкової величини ξ визначається за формулою:

$$D(\xi)=\int_{-\infty}^{-\infty}\left[x-M(\xi)\right]p_{\xi}(x)dx.$$

Або як і в дискретному випадку:

$$D(\xi)=M\left(\xi^2\right)-M(\xi)^2,$$

де $M(\xi)^2=\int_{-\infty}^{-\infty}x^2 p_{\xi}(x)dx$.

Властивості дисперсії:

1. $D(\xi)=M\left(\xi^2\right)-M(\xi)^2$.
2. $D(C)=C,\quad C=const$.
3. $D(C\xi)=C^2D(\xi),\quad C=const$.
4. $D(C+\xi)=D(\xi),\quad C=const$.
5. $D(\xi_1+...+\xi_n)=\sum_{i=1}^{n}D(\xi_i)$, якщо $\xi_1,...,\xi_n$ попарно незалежні випадкові величини.

Дисперсія являється мірою розкиду значень випадкової величини навколо її математичного очікування.

В якості інших характеристик, які описують властивості розподілу неперервної випадкової величини, використовують початкові і центральні моменти.

Початковим моментом неперервної випадкової величини k-го порядку називається математичне очікування k-го степеню [14]:

$$\alpha_k=M\left(\xi^k\right).$$

Очевидно, що $M(\xi)=\alpha_1$.

Центральним моментом неперервної випадкової величини k-го порядку називається математичне очікування k-го степеню відхилення випадкової величини від її математичного очікування [14, 15]:

$$\mu_k = M\left(\left[\xi - M(\xi)\right]^k\right).$$

Очевидно, що $\mu_0 = 1$, $\mu_1 = 0$, $\mu_2 = D(\xi)$.

Якщо щільність розподілу випадкової величини симетрична відносно математичного очікування, то всі центральні моменти непарного порядку рівні нулю.

Величина $\beta_\xi = \mu_3 / \sigma^3$ називається коефіцієнтом асиметрії випадкової величини ξ. Він характеризує «скошеність» кривої розподілу відносно математичного сподівання. Для симетричних законів розподілу $\mu_3 = 0$, тому теж $\beta_\xi = 0$. Для розподілів скошених вліво $\beta_\xi < 0$, для скошених вправо – $\beta_\xi > 0$ (рис. 1.8, а) [16].

Величина $\gamma_\xi = \mu_4 / \sigma^4 - 3$ називається коефіцієнтом ексцесу випадкової величини ξ. Він характеризує «згладженість» або «крутизну» розподілу по відношенню до нормального розподілу. Так як для нормального розподілу $\mu_4 / \sigma^4 = 3$, то $\gamma_\xi = 0$. Для гостроверхих розподілів $\gamma_\xi > 0$, для сгладжених – $\gamma_\xi < 0$ (рис. 1.8, б) [16, 17].

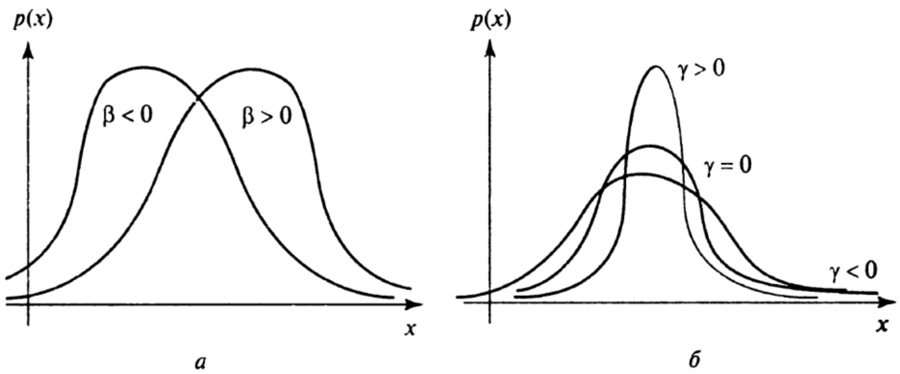


Рис. 1.8. Форми кривих розподілу при різних значеннях коефіцієнту асиметрії β_ξ (а) і коефіцієнту ексцесу γ_ξ (б)

Модою Mo випадкової величини називають її найбільш імовірне значення.

Для дискретної випадкової величини мода Mo визначається з ряду розподілу або за допомогою багатокутника розподілу (рис. 1.9, а). Для неперервної випадкової величини модою Mo називають точку локального максимуму функції щільності імовірності (рис. 1.9, б).

Якщо розподіл має один максимум, то випадкова величина називається унімодальною, більше одного – мультимаодальною. Розподіл, який має мінімум, називається антимодальним.

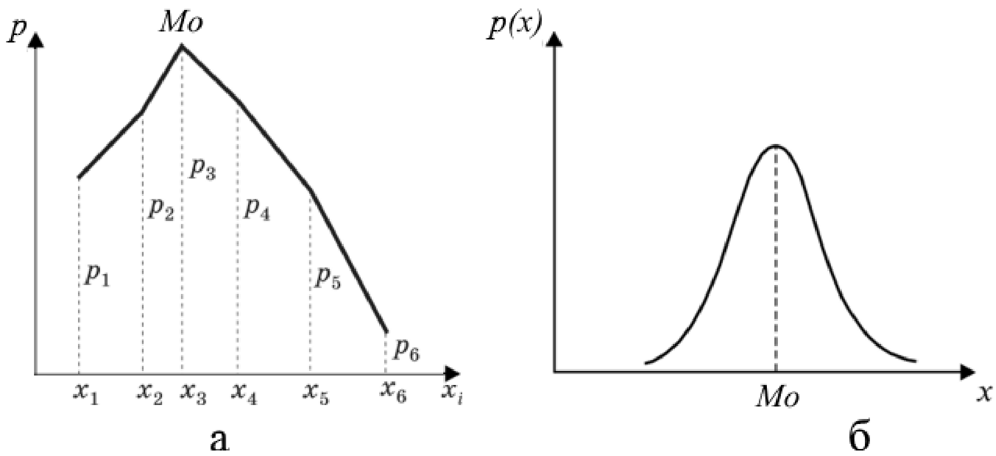


Рис. 1.9. Приклад графічного визначення моди *Me* дискретної (а) і неперервної випадкової величини (б)

Медіаною неперервної випадкової величини є таке значення Me, що:

$$P\left(\xi < Mo(\xi)\right) = P\left(\xi > Mo(\xi)\right) = 0.5.$$

Для дискретної випадкової величини ξ з непарною кількістю можливих значень мода визначається як середній елемент упорядкованої множини можливих значень даної випадкової величини. Якщо ж кількість елементів дискретної випадкової величини ξ парна, то мода визначається як будь-яке значення, що лежить між сусідніми елементами, що знаходяться посередині. Також часто за медіану у випадку парної кількості можливих значень приймають середнє значення сусідніх елементів [18].

Геометрично медіана ділить криву розподілу на дві частини, які мають однакову площу під кривою (рис. 1.10).

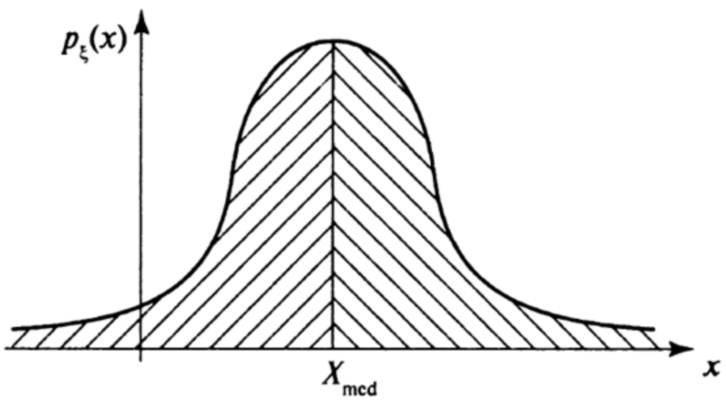


Рис. 1.10. Медіана неперервної випадкової величини

Математичне очікування, мода, медіана, початкові і центральні моменти являються основними числовими характеристиками випадкової величини.

Доволі часто необхідно оцінити ступінь подібності або схожості значень випадкових величин. Для цього використовують числові характеристики залежностей, серед яких коваріація та коефіцієнт кореляції.

Коваріацією двох випадкових величин ξ_1, ξ_2 називається число:

$$\operatorname{cov}(\xi_1, \xi_2) = M\left[\left(\xi_1 - M(\xi_1)\right)\left(\xi_2 - M(\xi_2)\right)\right].$$

за умови, що для випадкових величин ξ_1, ξ_2 існує математичне очікування.

Коваріація має наступні властивості:

1. Якщо ξ_1, ξ_2 незалежні випадкові величини, то $\operatorname{cov}(\xi_1, \xi_2) = 0$.
2. $\operatorname{cov}(\xi_1, \xi_2) = \operatorname{cov}(\xi_1, \xi_2)$.
3. $\operatorname{cov}(C\xi_1, \xi_2) = C\operatorname{cov}(\xi_1, \xi_2)$,
   $\operatorname{cov}(\xi_1, C\xi_2) = C\operatorname{cov}(\xi_1, \xi_2)$.
4. $\operatorname{cov}(\xi_1 + \xi_2, \xi_3) = \operatorname{cov}(\xi_1, \xi_3) + \operatorname{cov}(\xi_2, \xi_3)$,
   $\operatorname{cov}(\xi_1, \xi_2 + \xi_3) = \operatorname{cov}(\xi_1, \xi_2) + \operatorname{cov}(\xi_1, \xi_3)$.

5. $\operatorname{cov}(\xi_1,\xi_1)=D(\xi_1)$.

Коефіцієнтом кореляції двох випадкових величин ξ_1,ξ_2 називається число:

$$\rho=\frac{\operatorname{cov}(\xi_1,\xi_2)}{\sqrt{D(\xi_1)}\sqrt{D(\xi_2)}}=\frac{\operatorname{cov}(\xi_1,\xi_2)}{\sigma(\xi_1)\sigma(\xi_2)}.$$

Коефіцієнт кореляції являється однією з важливих мір залежності випадкових величин. З властивостей коваріації випливає, що коефіцієнт кореляції приймає значення $[-1,+1]$, тим самим відображуючи силу залежності (по абсолютному значенню)і характер залежності (позитивну або негативну) [19].

Коефіцієнт кореляції ρ характеризує так звану лінійну імовірнісну залежність двох випадкових величин, яка полягає в тому, що при зростанні однієї випадкової величини друга має тенденцію до лінійної зміни значення. Тобто з ростом ξ_1 значення випадкової величини ξ_2 теж зростає при $\rho>0$ і спадає при $\rho<0$. Тому при $\rho>0$ говорять, що величини ξ_1,ξ_2 знаходяться в позитивній кореляційній залежності, а при $\rho>0$ – в негативній кореляційній залежності [19].

Для незалежних випадкових величин коефіцієнт кореляції $\rho=0$. Такі випадкові величини називаються некорельованим. З незалежності випадкових величин слідує їх некорельованість, проте зворотне твердження не завжди вірне.

1.2.4. Багатовимірні випадкові величини

Особливу роль в теорії імовірностей відіграють багатовимірні випадкові величини (вектори), оскільки реальні явища доволі часто залежать від декількох випадкових величин.

Нехай на імовірнісному просторі $\{\Omega, F, P\}$ задано випадкові величини $\xi_1(\omega),...,\xi_n(\omega)$, де $\omega \in \Omega$. Подібний упорядкований набір називається багатовимірним випадковим вектором або n-вимірною випадковою величиною і позначається $\xi = (\xi_1,...,\xi_n)$. Самі величини ξ_i називаються складовими (компонентами) випадкового вектору [20].

Геометричною інтерпретацією багатовимірного випадкового вектора є випадкова точка з координатами $(\xi_1,...,\xi_n)$ або ж випадковий вектор направлений з початку координат в дану точку.

Для наочності і простоти всі наступні викладки наведені для двовимірного випадкового дискретного вектора, проте вони дійсні для випадкових величин будь-якої розмірності.

Сумісною функцією розподілу імовірності випадкового вектора (ξ_1, ξ_2) називається функція:

$$F(x, y) = P(\xi_1 < x, \xi_2 < y).$$

Окремими функціями розподілу називають функції розподілу складових ξ_1, ξ_2 – $F_{\xi_1}(x)$, $F_{\xi_2}(y)$.

Якщо випадкові величини ξ_1, ξ_2 незалежні, то сумісна функція розподілу рівна добутку функцій розподілу складових [20]:

$$F(x, y) = F_{\xi_1}(x) F_{\xi_2}(y).$$

Вірне і зворотне – якщо $F_{\xi_1}(x) F_{\xi_2}(y) = F(x, y)$, то ξ_1, ξ_2 – незалежні випадкові величини.

Багатовимірні функції розподілу володіють аналогічними властивостями, що й одновимірні:

1. Для будь-якого $x \in R$ справедлива нерівність $0 \le F(x_i,...,x_n) \le 1$.
2. Функція розподілу – неспадна функція по кожному з аргументів
3. Справедливі наступні граничні умови:

$$F(+\infty,...,+\infty) = 1,$$

$$F(x_1,...,x_{n-1},-\infty)=...=F(-\infty,x_2,...,x_n)=0.$$

4. Функція розподілу неперервна зліва по кожному з аргументів.

Геометрична інтерпретація функції розподілу $F(x,y)$ визначає імовірність потрапляння випадкового вектора (ξ_1,ξ_2) в нескінченний квадрат $(\xi_1<x,\xi_2<y)$ з вершиною в точці (x,y) (рис. 1.11). Дане визначення вірне як дискретного, так і неперервного випадкового вектора.

Дана геометрична інтерпретація дозволяє дати наочне пояснення граничним властивостям функції багатовимірного розподілу. Якщо $x\to-\infty$ (або $y\to-\infty$), то права межа (або верхня межа) нескінченного квадрату необмежено зміщується ліворуч (або вниз) і імовірність потрапляння в даний квадрат прагне до нуля. При $x\to+\infty$ і $y\to+\infty$ даний квадрат перетворюється у всю площину XOY і потрапляння випадкової точки в дану площину являється достовірною подією.

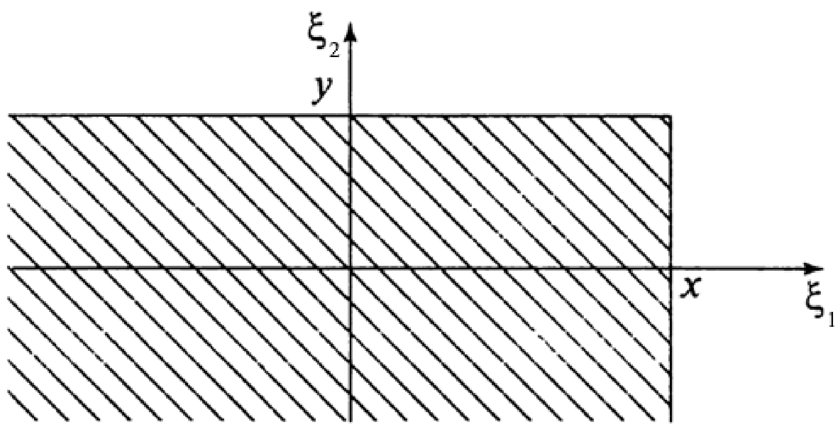


Рис. 1.11. Геометрична інтерпретація двовимірної функції розподілу

Якщо ж один з аргументів рівний $+\infty$, то функція розподілу випадкового вектора перетворюється в функцію розподілу випадкової величини, яка відповідає даному аргументу (рис. 1.12):

$$F(x,+\infty)=F_{\xi_1}(x),$$
$$F(+\infty,y)=F_{\xi_2}(y).$$

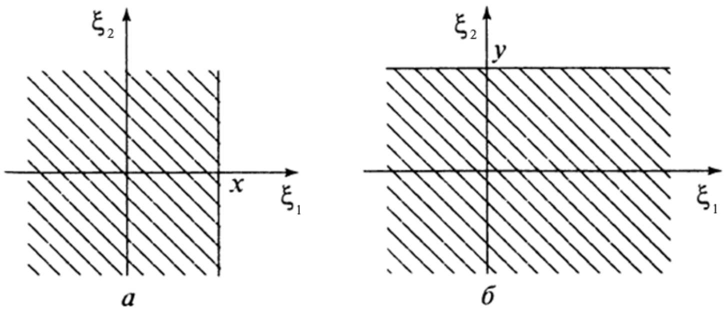


Рис. 1.12. Отримання функцій розподілу компонентів $F_{\xi_1}(x)$ (а) і $F_{\xi_2}(y)$ (б) з сумісної функції розподілу

Аналогічно випадковим величинам багатовимірні випадкові величини можуть бути дискретними, неперервними і абсолютно неперервними.

Якщо імовірнісний простір, на якому випадковий вектор заданий, то даний вектор – дискретний.

Випадковий вектор називається неперервним, якщо його сумісна функція розподілу неперервна. Випадковий вектор називається абсолютно неперервним, якщо існує невід'ємна функція $p_{\xi_1,\xi_2}(x,y)$, яка називається сумісною щільністю розподілу випадкових величин ξ_1,ξ_2 (або вектора) і задовольняє рівність [20, 21]:

$$F(x,y)=\int_{-\infty}^{x}\int_{-\infty}^{y}p_{\xi_1,\xi_2}(x,y)dydx .$$

Будь-яке співвідношення між можливими значеннями випадкового вектора і їх імовірностями називається сумісним законом розподілу.

Сумісний закон розподілу імовірностей дискретних величин (ξ_1,ξ_2)задається набором імовірностей p_{ij} одночасного настання подій $(\xi_1=x_i,\xi_2=y_j)$ (табл. 2).

Окремим законом розподілу випадкової величини ξ_1 називається набір імовірностей події $(\xi_1=x_i)$. Якщо заданий сумісний закон розподілу випадкових величин (ξ_1,ξ_2), то окремий закон розподілу ξ_1 можна знайти за допомогою формули [22]:

$$P_{\xi_1}(x_i) = P(\xi_1 = x_i) = \sum_j P(\xi_1 = x_i, \xi_2 = y_j) = \sum_j p_{ij}\ .$$

Таблиця 1.2. Сумісний закон розподілу

$\xi_2 = y_j$ / $\xi_1 = x_i$	y_1	y_2	...	y_m
x_1	p_{11}	p_{12}	...	p_{1m}
x_2	p_{21}	p_{22}	...	p_{2m}
...	...	...	...	...
x_n	p_{n1}	p_{n2}	...	p_{nm}

Дійсно, подія $(\xi_1 = x_i)$ може відбутися з однією із подій $(\xi_1 = x_i, \xi_2 = y_1),...,(\xi_1 = x_i, \xi_2 = y_n)$, які є несумісними і їх сума є подією $\xi_1 = x_i$. Тобто $(\xi_1 = x_i) = \bigcup_{j=1}^{m} (\xi_1 = x_i, \xi_2 = y_1)$, звідки слідує $P_{\xi}(x_i) = \sum_{j=1}^{m} p_{ij}$.

Аналогічним чином знаходяться окремі закони розподілу інших випадкових величин у дискретних випадкових векторах.

Законом розподілу для неперервного випадкового вектора виступає сумісна щільність розподілу $p_{\xi_1,\xi_2}(x,y)$, яка визначається як:

$$p(x,y) = F_{xy}''(x,y).$$

Сумісна функція розподілу володіє наступними властивостями:

1. Функція додатна – $p(x,y) \geq 0$.
2. $\int\limits_{-\infty}^{+\infty}\int\limits_{-\infty}^{+\infty} p(x,y)dxdy = 1$.

Якщо задана сумісна щільність розподілу $p_{\xi_1,\xi_2}(x,y)$ неперервного вектора (ξ_1,ξ_2), то щільності складових ξ_1,ξ_2 називаються окремими щільностями і визначаються за формулами [22]:

$$p_{\xi_1}(x) = \int_{-\infty}^{+\infty} p(x, y) dy,$$

$$p_{\xi_2}(y) = \int_{-\infty}^{+\infty} p(x, y) dx.$$

Таким чином, розподіл кожної випадкової величини можна відновити по відомому сумісному закону розподілу імовірностей (у випадку дискретного вектора) або щільності імовірностей (у випадку неперервного вектора).

Випадкові дискретні величини ξ_1, ξ_2 називаються незалежними, якщо для будь-яких множин A і B виконується умова:

$$P(\xi_1 \in A, \xi_2 \in B) = P(\xi_1 \in A) P(\xi_2 \in B).$$

Для неперервних випадкових ξ_1, ξ_2 величин зі щільностями $p_{\xi_1}(x)$ і $p_{\xi_2}(y)$ незалежність означає те, що при будь-яких x і y виконується рівність:

$$p_{\xi_1, \xi_1}(x, y) = p_{\xi_1}(x) p_{\xi_2}(y).$$

Проте для моделювання реальних випадкових процесів випадкові величини підходять лише за умови значного спрощення досліджуваних явищ. Випадкові процеси, які є узагальненням випадкових величин, дозволяють враховувати зміну випадкового процесу з плином часу (еволюцію), що дозволяє використовувати для симуляції реальних випадкових явищ та процесів.

1.3. Випадкові процеси

Теорія випадкових процесів (випадкових функцій) – це розділ математичної науки, що вивчає закономірності випадкових явищ в динаміці їх розвитку.

При вивченні різних реальних явищ розглядають різноманітні величини, що залежать від часу $x = x(t)$. Такі величини прийнято називати процесами. Говорячи про випадкові процеси, як правило, мають на увазі деяку випадкову величину, яка залежить від часу $\xi(t)$. Будемо називати випадковим процессом $\xi(t)$ функцію від дійсного параметра $t \in T$ (часу), значення якої при кожному t є випадковою величиною.

Строге визначення випадкового процесу полягає в наступному. Набір випадкових величин $\xi(t) = \left(\xi(t,\omega), \quad t \in T, \omega \in \Omega\right)$, визначених на одному і тому ж імовірнісний просторі (Ω, F, P), називається випадковою функцією [23].

Нехай задано імовірнісний простір (Ω, F, P) і T – деяка множина моментів часу. І нехай кожній елементарній події $\omega \in \Omega$ поставлена у відповідність функція $\xi_t = \xi(t) = \xi(t,\omega)$, $t \in T$, зі значеннями в n-мірному просторі ($n \geq 1$) така, що при кожному фіксованому $t \in T$ функція $\xi(t,\omega)$ є випадковою величиною. Ця функція називається випадковим процессом [23, 24].

Відповідно, випадковий процес – це сукупність випадкових величин, що залежать від часу. Як випливає з визначення, $\xi(t,\omega)$ є n-мірною випадковою величиною при будь-якому фіксованому $t \in T$.

При фіксованому $\omega \in \Omega$ функція $\xi(t,\omega)$ перетворюється в $\xi(t)$ і називається реалізацією (траєкторією або вибірковою функцією) випадкового процесу. Сукупність всіх реалізацій випадкового процесу називається ансамблем (сімейством) реалізацій. Дана функція являється повністю детермінованою.

При фіксованому t для випадкового процесу $\xi(t,\omega)$ отримуємо випадкову величину, яку називають перетином випадкового процесу в

момент часу t. В подальшому випадковий процес $\xi(t,\omega)$ для зручності буде позначатися $\xi(t)$.

Таким чином, випадковий процес $\xi(t,\omega)$ поєднує в собі риси випадкової величини і функції. Якщо зафіксувати значення аргументу t, то випадковий процес перетворюється у звичайну випадкову величину, якщо зафіксувати ω, то в результаті кожного випробування він перетворюється на звичайну невипадкову функцію [25].

1.3.1. Класифікація випадкових процесів

Випадковий процес $\xi(t)$, $t \in T$, у якого множина значень T не більше ніж злічення, називається процесом з дискретним часом або випадковою послідовністю (рис.1.13, а, в). Якщо множина T є проміжком деякої прямої, тобто являється незліченною множиною, то $\xi(t)$ називається процесом з безперервним часом (рис.1.13, б, г). Якщо значенням випадкового процесу являється неперервною випадковою величиною, то такий процес називається процесом з неперервними станами (рис. 1.13, в, г), якщо значення випадкового процесу – дискретна випадкова величина, то процес з дискретною станами (рис. 1.13, а, б) [25, 26].

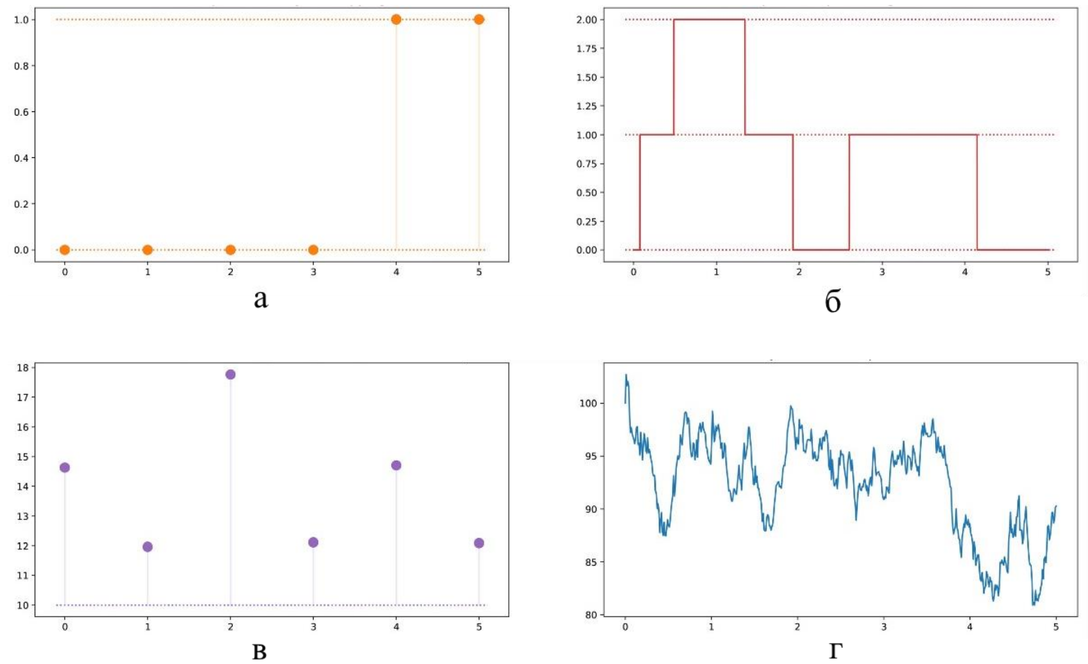


Рис. 1.13. Дискретний в часі і по значенням (а), неперервний в часі і дискретний по значенням (б), дискретний у часі і неперервний по значенням (в), неперервний в часі і по значенням (г) випадковий процес

Окрім класифікації випадкових процесів по неперервності в часі і по значенням існує класифікація по видам стаціонарності (рис. 1.14) [27-30].

Важливим класом випадкових процесів являються стаціонарні випадкові процеси. Випадковий процес називається стаціонарним, якщо всі багатовимірні закони розподілу залежать тільки від взаємного розташування моментів часу $t_1, t_2, ..., t_n$, але не від самих значень цих величин. Іншими словами, випадковий процес називається стаціонарним, якщо його імовірнісні закономірності незмінні в часі. В іншому випадку, він називається нестаціонарним.

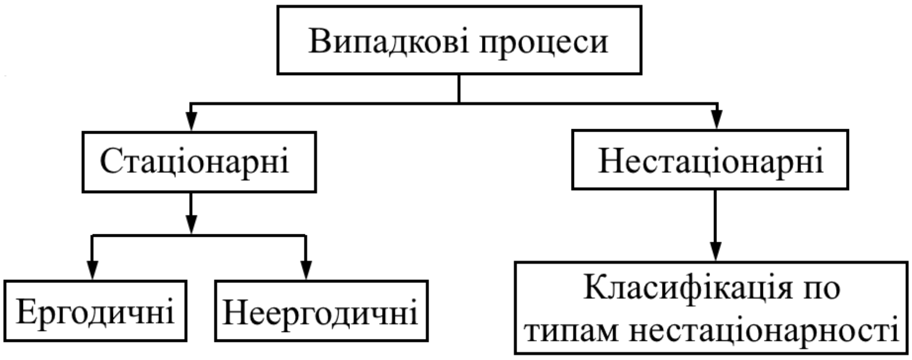


Рис. 1.14. Класифікація випадкових процесів по випадам стаціонарності

Якщо при визначенні моментних функцій (числових характеристик і характеристик залежностпей випадкової величини) стаціонарного випадкового процесу операцію усереднення по ансамблю можна замінити усередненням за часом, то такий стаціонарний випадковий процес називається ергодичним. В іншому випадку випадковий процес являється неергодичним [31].

Нестаціонарні випадкові процеси не набули великого поширення, тому їх класифікація не наводиться.

1.3.2. Функція розподілу і щільність імовірності

Розглянемо перетин випадкового процесу $\xi(t)$ в момент t_1. Функція $F(x_1,t_1)=P(\xi(t_1)\le x_1)$ носить назву одновимірної функції розподілу випадкового процесу в момент часу t_1. Якщо зафіксувати два моменти часу t_1 і t_2, то ймовірність спільного виконання випадкових подій $\xi(t_1)\le x_1$ і $\xi(t_2)\le x_2$ задає двовимірну функцію розподілу випадкового процесу

$$F(x_1,t_1;x_2,t_2)=P(\xi(t_1)\le x_1,\xi(t_2)\le x_2).$$

Нехай $t_1<t_2<...<t_n$ – деякі фіксовані моменти часу. Розглянемо n-мірну випадковий величину $(\xi(t_1),\xi(t_2),...,\xi(t_n))$. Розподіл ймовірностей цієї випадкової величини називається n-мірною функцією розподілу випадкового процесу, а функція розподілу цієї випадково величини відповідно n-мірну функцію розподілу випадкового процесу:

$$F(x_1,t_1;...;x_n,t_n)=P(\xi(t_1)\le x_1,...,\xi(t_n)\le x_n).$$

Варто зауважити, що n-мірна функція розподілу випадкового процесу є функцією $2n$ змінних $x_1,...,x_n,t_1,...,t_n$, коли функція розподілу n-мірної випадкової величини є функцією $2n$ змінних $x_1,...,x_n$.

Також для функції розподілу $F\left(x_1,t_1;...;x_n,t_n\right)=P\left(\xi\left(t_1\right)\le x_1,...,\xi\left(t_n\right)\le x_n\right)$ повинні виконуватися умови узгодженості [32]:

$$F\left(x_1,t_1;x_2,t_2...;x_m,t_m;+\infty,t_{m+1};...;+\infty,t_n\right)=$$
$$=F\left(x_1,t_1;x_2,t_2...;x_m,t_m\right),\quad \forall m<n.$$

$$F\left(x_{i_1},t_{i_1};x_{i_2},t_{i_2}...;x_{i_n},t_{i_n}\right)=F\left(x_1,t_1;x_2,t_2...;x_n,t_n\right),$$

де $i_1,i_2,...,i_n$ – будь-яка перестановка індексів $1,2,...,n$.

Якщо існує така невід'ємна функція $p\left(x_1,t_1;...;x_n,t_n\right)$, що для всіх x виконується умова

$$F\left(x_1,t_1;...;x_n,t_n\right)=\int_{-\infty}^{x_1}...\int_{-\infty}^{x_n}p\left(x_1,t_1;...;x_n,t_n\right)dx_1...dx_n,$$

то вона називається n-мірної щільністю розподілу випадкового процесу.

В точках неперервності щільність розподілу знаходиться за формулою:

$$p\left(x_1,t_1;...;x_n,t_n\right)=\frac{\partial^n F\left(x_1,t_1;...;x_n,t_n\right)}{\partial x_1...\partial x_n}.$$

При цьому умова узгодженості приймає вигляд:

$$p\left(x_1,t_1;x_2,t_2...;x_m,t_m\right)=\int_{-\infty}^{x_1}\int_{-\infty}^{x_2}...p\left(x_1,t_1;x_2,t_2...;x_m,t_m\right)dx_{m+1}...dx_n.$$

На рис. 1.15 зображено декілька реалізацій деякого випадкового процесу. Нехай перетин даного процесу при заданому значенні t являється неперервною випадковою величиною. Тоді випадковий процес $\xi\left(t\right)$ при даному t визначається щільністю імовірності $p\left(x,t\right)$.

Проте щільністю імовірності $p\left(x,t\right)$ не являється вичерпним описанням випадкового процесу $\xi\left(t\right)$, оскільки вона не описує залежність між перетинами в різні моменти часу

Випадковий процес $\xi\left(t\right)$ представляє собою сукупність всіх перетинів при всіх можливих значеннях t, тому для його описання необхідно

розглядати багатовимірну випадкову величину $\left(\xi(t_1),\xi(t_2),...,\xi(t_n)\right)$, яка складається з усіх перетинів даного процесу. Взагалі таких перетинів у випадку неперервного випадкового процесу безліч, але для його описання часто вдається обійтися скінченною кількістю перетинів.

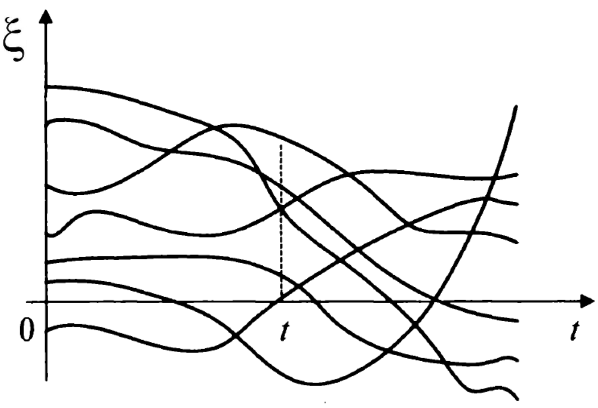


Рис. 1.15. Приклади реалізацій випадкового процесу

Говорять, що випадковий процес має порядок n, якщо він повністю визначається щільністю сумісного розподілу $p\left(x_1,t_1;x_2,t_2...;x_n,t_n\right)$ n випадкових перетинів процесу, тобто щільністю n-мірної випадкової величини $\left(\xi(t_1),\xi(t_2),...,\xi(t_n)\right)$, де $\xi(t_i)$ – перетин випадкового процесу $\xi(t)$ в момент часу t_i, де $i=1,2,...,n$.

1.3.3. Числові характеристики і характеристики залежностей

Як і випадкова величина, випадковий процес може бути описаний числовими характеристиками. Для випадкової величини дані характеристики являються постійними числами, а для випадкового процесу – невипадковими функціями [33].

Математичним очікуванням випадкового процесу $\xi(t)$ називається невипадкова функція $m_\xi(t)$, яка при будь-якому значенні t рівна

математичному очікуванню відповідного перетину випадкового процесу $\xi(t)$:

$$m_{\xi}(t)=M\left(\xi(t)\right)=\int_{-\infty}^{+\infty} x dF(x,t).$$

Математичне очікування визначається одномірною функцією розподілу $F(x,t)$.

Дисперсією випадкового процесу $\xi(t)$ називається невипадкова функція $D_{\xi}(t)$, яка при будь-якому значенні t рівна дисперсії відповідного перетину випадкового процесу $\xi(t)$:

$$D_{\xi}(t)=M\left[\left(\xi(t)-m_{\xi}(t)\right)^{2}\right]=\int_{-\infty}^{+\infty}\left[x-m_{\xi}(t)\right]dF(x,t).$$

Середньоквадратичним відхиленням $\sigma_{\xi}(t)$ випадкового процесу $\xi(t)$ називається наступна невипадкова функція:

$$\sigma_{\xi}(t)=\sqrt{D_{\xi}(t)}.$$

Математичне очікування випадкового процесу характеризує середню траєкторію всіх можливих його реалізацій, а дисперсія – розкид реалізацій відносно середньої траєкторії.

Наведених вище характеристик не достатньо для однозначного визначення процесу, так як вони визначаються лише одномірним законом розподілу. На рис 1.16 зображені випадкові процеси з приблизно однаковими математичними очікуваннями і дисперсіями [34].

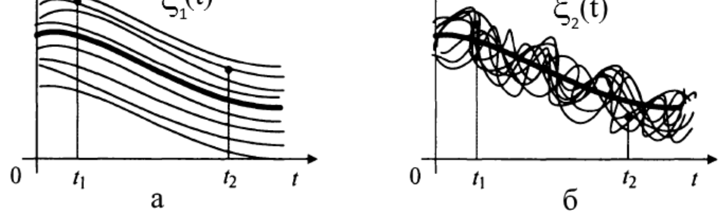


Рис. 1.16. Випадкові процеси з однаковими числовими характеристиками і різними характеристиками залежностей

Процес $\xi_1(t)$ (рис. 1.16, а) характеризується тісною імовірнісною залежністю між перетинами $\xi_1(t_1)$ і $\xi_1(t_2)$, тоді як для процесу $\xi_2(t)$ (рис. 1.16, б) подібна залежність майже відсутня. Щоб врахувати подібну залежність використовують коваріаційну функцію випадкового процесу, яка рівна коваріації відповідних перетинів $\xi(t_1)$, $\xi(t_1)$ випадкового процесу $\xi(t)$ в моменти часу t_1, t_2:

$$K_\xi(t_1,t_2) = \mathrm{cov}\left(\xi(t_1)\xi(t_2)\right) = M\left[\left(\xi(t_1) - m_\xi(t_1)\right)\left(\xi(t_2) - m_\xi(t_2)\right)\right].$$

При $t_1 = t_2$ коваріаційна функція дорівнює дисперсії випадкового процесу $\xi(t)$.

Коваріаційна функція $K_\xi(t_1,t_2)$ характеризує ступінь лінійної залежності між перетинами випадкового процесу і ступінь розкиду даних перетинів відносно математичного очікування.

Тому також розглядають нормовану коваріаційну функцію (функцію кореляції) [35]:

$$\rho_\xi(t_1,t_2) = \frac{K_\xi(t_1,t_2)}{\sigma_\xi(t_1)\sigma_\xi(t_2)}.$$

Дана функція характеризує ступінь лінійної залежності між перетинами випадкового процесу $\xi(t)$.

Для двох випадкових процесів $\xi(t)$ і $\eta(t)$ вводиться поняття взаємної функції кореляції (функції крос-кореляції) [35, 36]:

$$R_{\xi\eta}(t_1,t_2) = M\left[\xi(t_1)\eta(t_2)\right],$$

а сумісна кореляційна функція двох випадкових процесів визначається як матрична функція [37]:

$$\begin{pmatrix} R_\xi(t_1,t_2) & R_{\xi\eta}(t_1,t_2) \\ R_{\eta\xi}(t_1,t_2) & R_\eta(t_1,t_2) \end{pmatrix}.$$

Наведені вище характеристики випадкових процесів мають ті самі властивості випадкових величин, оскільки являються їх узагальненням.

Прикладами випадкових процесів є гаусівський процес, пуасонівський процес, авторегресійні моделі, ланцюги Маркова, випадкове блукання, процес Вінера.

1.4. Марковська властивість. Марковські процеси

Одна з властивостей, яка сильно спрощує дослідження випадкових процесів – це Марковська властивість. Майбутній стан випадкових процесів, які володіють цією властивістю, не залежать від попередніх станів процесу і залежить лише від теперішнього стану. Процеси, які володіють даною властивістю, називаються марковськими [38].

Марковські випадкові процеси (процеси без наслідку, процеси без пам'яті), є зручною математичною моделлю для дослідження багатьох реальних процесів.

Нехай система може перебувати в різних станах і її еволюція в часі носить стохастичний характер, тобто стан системи в момент часу t в загальному випадку не визначається однозначно станами системи в попередні моменти $s<t$. Дана властивість називається фільтрацією. Тоді стан цієї системи можна описати деяким випадковим процесом $\xi(t)$, заданим на інтервалі часу T і який приймає значення з множини X [39-41].

Нехай задані n перетинів даного випадкового процесу $\xi(t_1),\xi(t_2),...,\xi(t_{n+1})$ в моменти часу $t_1<t_2<...<t_{n+1}$. Тоді випадковий процес $\xi(t)$ називається марковським або таким, що володіє марковською властивістю, якщо умовна щільність розподілу ймовірностей:

$$p(x_{n+1},t_{n+1} \mid x_1,t_1;x_2,t_2;...;x_n,t_n)=\frac{p(x_1,t_1;x_2,t_2;...;x_{n+1},t_{n+1})}{p(x_1,t_1;x_2,t_2;...;x_n,t_n)}.$$

не залежить від значень процесу в моменти $t_1,t_2,...,t_{n-1}$, а визначається лише станом процесу $\xi(t_n)=x_n$. Що еквівалентно:

$$p(x_{n+1},t_{n+1} \mid x_1,t_1;x_2,t_2;...;x_n,t_n)=p(x_{n+1},t_{n+1};x_n,t_n).$$

Дану умовну щільність імовірностей називають імовірністю переходу системи зі стану x_n, в якому вона знаходилась в момент часу t_n, в стан x_{n+1} в момент часу $t_{n+1} > t_n$.

Звідси випливає, що початковий розподіл $p(x_1, t_1)$ і імовірність переходу $p(x,t;y,x)$, де $t > s$ повністю задають марковський процес.

Марковські процеси можна класифікувати в залежності від структури множини значень випадкового процесу X й інтервалу спостереження T (табл. 1.3) [42].

Якщо множина значень X зліченна або скінченна (дискретна), то марковський процес називається ланцюгом Маркова. Множину значень ланцюга Маркова X називають фазовим простором або простором станів ланцюга Маркова.

Ланцюг Маркова, у якого множина значень часу T зліченна або скінченна (дискретна), називається ланцюгом з дискретним часом, а ланцюг Маркова, у якого множина T – незліченна множина, називається ланцюгом з неперервним часом.

Таблиця 1.3. Класифікація процесу Маркова

	Дискретна множина значень	Неперервна множина значень
Дискретний час	Ланцюг Маркова з дискретним часом	Ланцюг Харісса, випадкове блукання
Безперервний час	Ланцюг Маркова з неперервним часом	Будь-який неперервний по часу і значенням стохастичний процес з марківською властивостю (броунівський рук)

1.5. Випадкове блукання та його моделі

Випадкове блукання – випадковий процес, який описує шлях, який складається з послідовності випадкових кроків на деякому математичному просторі. Являється найпростішою і найбільш вивченою математичною моделлю випадкового процесу з марковською властивістю [43].

Випадкове блукання являється марковським ланцюгом з дискретним часом і дискретною або неперервною множиною значень (броунівський рух). Також виділяють випадкові блукання різної розмірності.

В силу своєї простоти дана модель використовується в різних сферах в математиці, фізиці, економіці для моделювання реальних процесів з деякими наближеннями.

Дискретне в часі випадкове блукання описується наступною формулою:

$$Y_n = Y_0 + \sum_{i=1}^{n} Y_i ,$$

$$Y_i = \left(Y_0, Y_1, \ldots, Y_k\right),$$

де Y_n – значення випадкового блукання на n-тому кроці (в n-й момент часу); Y_0 – початкове значення випадкового блукання, Y_i – випадковий процес розмірності k, який визначає значення кроку випадкового блукання на i-тому кроці (в i-й момент часу), причому $i < n$ [44, 45].

Випадкове блукання називають дискретним, якщо множина значень випадкового блукання становить зліченну або скінченну множину. У випадку незліченної випадкової множини випадкове блукання називають неперервним.

Випадкове блукання називають k-мірним, якщо випадковий процес, який визначає значення кроку, складається з k випадкових величин.

Також випадкове блукання класифікують по виду законону розподілу випадкових величин процесу. Наприклад, нормальне гаусівське випадкове блукання ($\mu = 0, \sigma = 1$).

Випадкове блукання повністю описується параметрами випадкових процесів, які визначають значення кроку, а саме – математичним очікуванням і дисперсією.

Серед найбільш поширених моделей випадкового блукання [46-49]:

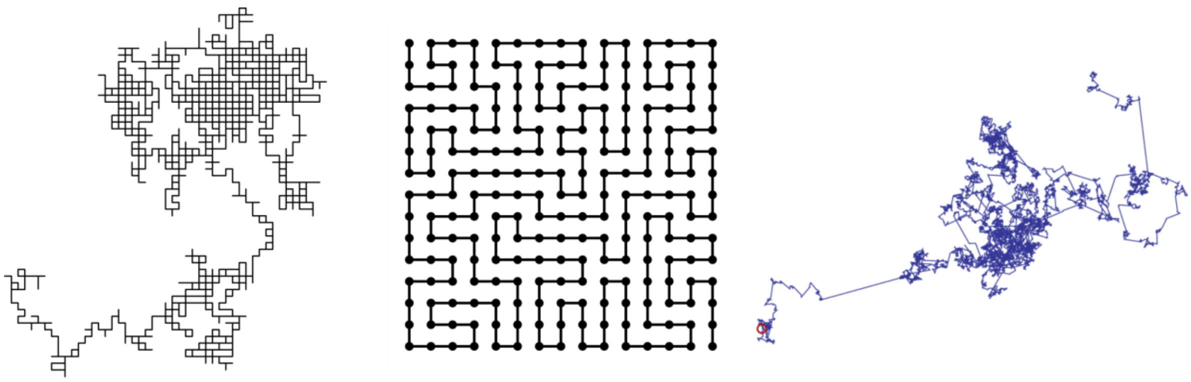
1. Випадкове блукання по решітці (рис. 12, а, б). Множина значень величини кроку $\{+1, -1\}$.

2. Випадкове блукання без повторень (рис. 12, б). Випадкове блукання не перетинає свою траєкторію.

3. Випадкове блукання levy flight (рис. 12, в). Випадкові величини випадкового процесу мають неперервний закон розподілу і коефіцієнт ексцесу $\gamma < 0$.

4. Нестаціонарні випадкові блукання (розподіл випадкових величин процесу залежить від часу).

5. Анізотропні k-мірні випадкові блукання (випадкові величини процесу мають різні закони розподілу).



а б в

Рис. 1.17 Двовимірне випадкове блукання на решітці (а), двовимірне випадкове блукання на решітці без повторень (б), випадкове блукання levy flight (в)

1.6. Ортогональні ряди для моделювання випадкового блукання

Ортогональні ряди мають нульове середнє значення функцій на інтервалі визначення, що дозволить при розробці нової моделі випадкового блукання не вносити зміни в початковий закон розподілу випадкових величин зміни і при цьому збільшити кількість проміжних станів без суттєвих затрат на обчислення і втрат швидкодії.

Серед найпоширеніших ортогональних рядів:

1. Різноманітні ряди Фур’є.
2. Функції Франкліна.
3. Функції Уолша.
4. Вейвлети Хаара та інші ортогональні вейвлети.

Особливе місце в обробці цифрових сигналів посідають функції Уолша. Вони використовуються в обробці цифрових зображень, розпізнаванні мови, в інформаційних каналах для скорочення надмірності при передачі інформації, що забезпечує зменшення обсягів інформації, яка передається.

При розробці моделей випадкового блукання особливо важливим являється відносно легка можливість комп’ютерної реалізації ортогональних рядів. З перелічених вище рядів функцій, такою властивістю володіють функції Уолша і ряди Фурье. Проте для визначення матриці перетворення ряду Фурє необхідно значна кількість обчислень, тоді як подібна матриця для Функцій складається з фіксованих значень (+1, –1) і знаходиться за рекурентною формулою, що дозволить значно пришвидшити розрахунки.

Висновки до першого розділу

Реальні випадкові процеси та явища достатньо точно описуються за допомогою марковських випадкових процесів. Найбільш дослідженими і

поширеними серед них являється випадкове блукання. Випадкове блукання використовується в таких галузях науки як фізика, економіка, біологія.

Для збільшення кількості проміжних точок випадкового блукання доцільно використовувати ортогональні ряди в силу нульової постійної складової. Для створення моделі випадкового блукання зі збільшеною кількістю проміжних точок було обрано функції Уолша, комп’ютерна реалізація яких потребує найменшу кількість обчислювальних потужностей.

Відновлювані джерела енергії характеризуються значною мінливістю і непередбачуваністю. Тому для забезпечення можливості точного моделювання необхідно враховувати стохастичний характер параметрів відновлюваних джерел. Розглянуті в даному розділі теоретичні матеріали достатньо точно здатні описувати подібні процеси.

У зв’язку з цим виникає задача створення математичної моделі, яка використовуватиметься для потреб моделювання випадкової складової відновлюваних джерел енергії.

2. МАТЕМАТИЧНА МОДЕЛЬ ВИПАДКОВОГО БЛУКАННЯ З ВИКОРИСТАННЯМ ОРТОГОНАЛЬНИХ РЯДІВ

2.1. Система функцій Уолша та її характеристики

Функції Уолша є узагальненням ортогонального ряду функцій Радемахера. Останні, в свою чергу, можуть визначатися трьома способами, проте найбільш простим і зрозумілим є визначення через тригонометричні функції [43]:

$$r_n(x) = sign[\sin(2^n \pi x)]. \tag{2.1}$$

де $n = 0,1,2,...$, $x = t/T$ – безрозмірний час, T – період функції, і $0 \le x < 1$, $sign$ – сигнум-функція.

Символом *sign* позначається сигнум-функція:

$$sign(x) = \begin{cases} 1, & x > 0, \\ -1, & x < 0. \end{cases}$$

Виходячи з 2.1, стає очевидно, що функції Радемахера визначені на додатній числовій множині $t \ge 0$, а множина значень дискретна і набуває значень $\{-1,+1\}$. На рис. 2.1 наведена система функцій Радемахера при $N = 4$.

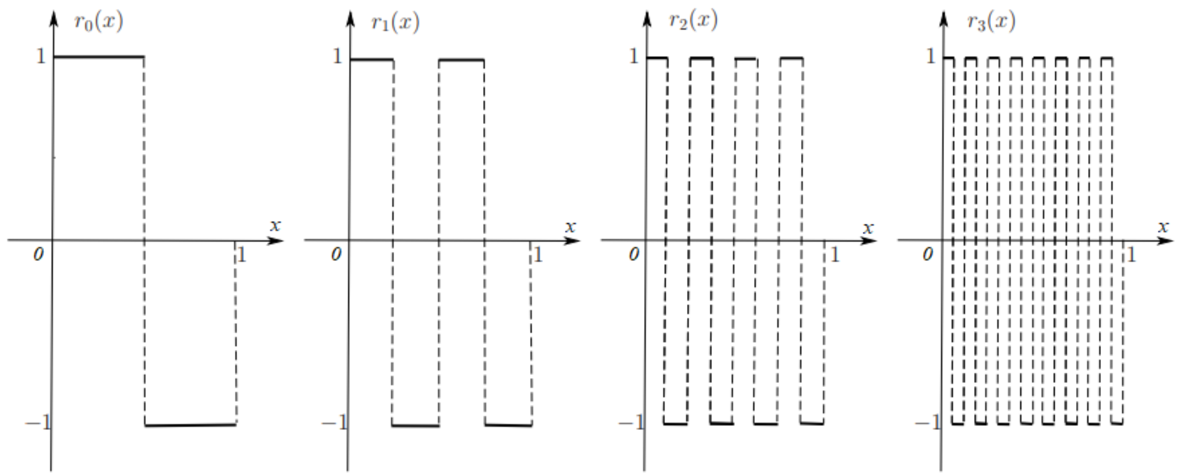


Рис. 2.1. Графіки перших чотирьох функцій Радемахера

Функції Уолша – система ортонормованих функцій, визначених на додатній числовій множині $t \geq 0$, множина значень яких дискретна і набуває значень $\{-1,+1\}$.

Система функцій Уолша класифікується по розмірності (довжина сигналу), яка визначається виразом:

$$N = 2^m,$$

де m – рівень сигналу; N, m – натуральні числа.

Також розрізняють різні способи упорядкування функцій, які її утворюють, всередині системи. Розрізняють наступні способи упорядкування [43, 44]:

1. По Уолшу.
2. По Пелі.
3. По Адамару.

Інколи нумерацію по Уолшу називаю ще оригінальною нумерацією або упорядкуванням по кількості знакозмін.

В загальному випадку кожна функція системи Уолша являє собою добуток скінченної кількості функцій Радемахера. Функції Радемахера, добутком яких є функція Уолша, називаються композицією.

Нехай необхідно знайти композицію для функції Уолша під номером n. Для цього представимо номер функції n у двійковій системі:

$$n^{bin} = n_0 2^0 + n_1 2^1 + \ldots + n_{m-1} 2^{m-1} = \sum_{i=0}^{m-1} n_i 2^i,$$

де $n_i \in \{0,1\}$, $m = \log_2 N$ – рівень сигналу.

Номер функції Уолша в коді Грея визначається наступним співвідношенням:

$$n^{gray} = 2^0\left(n_0^{bin} \oplus n_1^{bin}\right) + 2^1\left(n_1^{bin} \oplus n_2^{bin}\right) + \ldots + 2^{m-1}\left(n_{m-1}^{bin} \oplus n_m^{bin}\right) =$$

$$= \sum_{i=0}^{m-1} 2^i \left(n_i^{bin} \oplus n_{i+1}^{bin}\right).$$

Тоді композиція для функції Уолша під номером n в оригінальній нумерації визначається наступним чином [44, 45]:

$$wal_n(x) = \prod_{i=0}^{m-1} r_i(x)^{n_i^{gray}},$$

де n_i^{gray} – значення i-го розряду номера функції Уолша n в коді Грея, m – рівень сигналу.

Враховуючи, що для функції Уолша під номером $n=0$ код Грея має вид $n^{gray}=000$, маємо $wal_0(x)=1$.

Тоді система функцій Уолша в оригінальній нумерації визначається наступним чином [44-46]:

$$wal_0(x) = 1;$$

$$wal_n(x) = \prod_{i=0}^{m-1} r_i(x)^{n_i^{gray}}.$$

Наведемо етапи отримання системи функцій Уолша розмірністю $N = 2^3 = 8$ (табл. 2.1).

Таблиця 2.1

Номер n функції Уолша	n^{bin}	n^{gray}	Формула функції Уолша
0	000	000	$wal_0(x)=1$
1	001	001	$wal_1(x)=r_0(x)$
2	010	011	$wal_2(x)=r_0(x)r_1(x)$
3	011	010	$wal_3(x)=r_1(x)$
4	100	110	$wal_4(x)=r_1(x)r_2(x)$
5	101	111	$wal_5(x)=r_0(x)r_1(x)r_2(x)$
6	110	101	$wal_6(x)=r_0(x)r_2(x)$
7	111	100	$wal_6(x)=r_2(x)$

На рис. 2.2. зображено систему функцій Уолша розмірністю $N=8$ в оригінальній нумерації, аналітичні вирази яких наведені в табл. 2.1. Як вже

була сказано раніше, розглянуті функції впорядковані за кількістю знакозмін, проте при використанні даних функцій в електронно-обчислювальних машинах здобули поширення інші способи впорядкування.

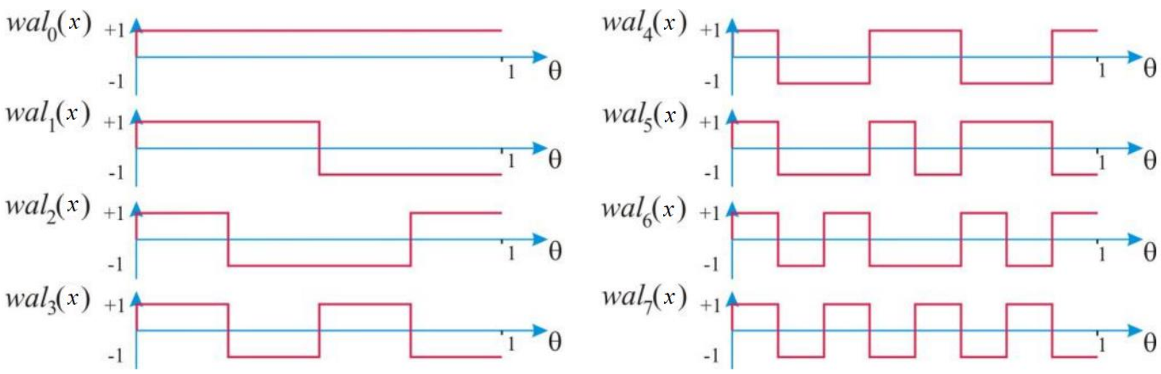


Рис. 2.2. Графіки функцій Уолша в упорядкуванні Уолша при $N=8$

В системі Пелі описані вище функції позначаються $pal_n(x)$. А порядок функцій визначається записом номера функції n у двійковому представленні. В системі Адамара $had_n(x)$ порядок визначається записом номера функції n у коді Грея в зворотному порядку. У табл. 2.2 наведено залежність між різними системами упорядкування при $N=2^3=8$ [47].

Таблиця 2.2

Число n	0	1	2	3	4	5	6	7
Число n в коді Грея	000	001	011	010	110	111	101	100
Номер n функції в системі Уолша	0	1	2	3	4	5	6	7
Номер n функції в системі Пелі	0	1	3	2	6	7	5	4
Номер n функції в системі Адамара	0	4	6	2	3	7	5	1

Система функцій Уолша, незалежно від способу впорядкування, має наступні властивості [48].

1. Функції Уолша ортонормовані:

$$\int_0^1 wal_i(x)wal_j(x)dx = \begin{cases} 1 & при \quad i = j, \\ 0 & при \quad i \neq j. \end{cases} \tag{2.2}$$

2. Математичне очікування функцій Уолша для всіх $i \neq 0$:

$$\int_0^1 wal_i(x)dx = 0, \quad i = 1, 2, ..., N-1.$$

3. Властивість мультиплікативності. Тобто добуток двох функцій Уолша є функцією Уолша тієї ж системи:

$$wal_i(x)wal_j(x) = wal_k(x),$$

де $k = i \oplus j$, $\oplus$ – порозрядне сума за модулем два. Дану властивість називають властивістю мультиплікативності.

4. Функція симетричним відносно середини інтервалу ($x = 0,5$) відповідають парні значення номеру функцій n і навпаки.

5. Система функцій Уолша має область визначення $[0,1]$. Поза даним інтервалом функції періодично повторюються.

Особливе місце в комп'ютерній реалізації системи функцій Уолша відіграють дискретні перетворення Уолша (ДПУ), в яких використовуються дискретні функції Уолша (ДФУ). Дані функції набули великого поширення при цифровій обробці сигналів. Також за рахунок використання рекурентних формул матрична реалізація являється особливо актуальною при використанні даних функцій в електронно обчислювальних машинах [49].

Дані дискретні функції являються дискретизованими по часу неперервними функціями Уолша, які визначаються за допомогою співвідношення [49, 50]:

$$W_n(k) = (-1)^{\sum_{i=0}^{m-1} n_i k_i} = \prod_{i=0}^{m-1} (-1)^{n_i k_i}, \tag{2.3}$$

де n – номер функції, $k = 0,1,2...,m-1$ – номер відліку, m – рівень сигналу, n_i – i-й розряд у двійковому записі номеру функції n, k_i – i-й розряд у двійковому записі номеру відліку k.

Приклад використання формули (2.3) наведено нижче для побудови ДФУ з номерами функцій $n = 2,7$ при $N = 2^3 = 8$.

$$2_{10} = 010_2, \qquad 7_{10} = 111_2,$$

$$W_2(0) = (-1)^{(0\cdot0+1\cdot0+0\cdot0)} = (-1)^0 = 1, \qquad W_7(0) = (-1)^{(1\cdot0+1\cdot0+1\cdot0)} = (-1)^0 = 1,$$

$$W_2(1) = (-1)^{(0\cdot1+1\cdot0+0\cdot0)} = (-1)^0 = 1, \qquad W_7(1) = (-1)^{(1\cdot1+1\cdot0+1\cdot0)} = (-1)^1 = -1,$$

$$W_2(2) = (-1)^{(0\cdot0+1\cdot1+0\cdot0)} = (-1)^1 = -1, \qquad W_7(2) = (-1)^{(1\cdot0+1\cdot1+1\cdot0)} = (-1)^1 = -1,$$

$$W_2(3) = (-1)^{(0\cdot1+1\cdot1+0\cdot0)} = (-1)^1 = -1, \qquad W_7(3) = (-1)^{(1\cdot1+1\cdot1+1\cdot0)} = (-1)^2 = 1,$$

$$W_2(4) = (-1)^{(0\cdot0+1\cdot0+0\cdot1)} = (-1)^0 = 1, \qquad W_7(4) = (-1)^{(1\cdot0+1\cdot0+1\cdot1)} = (-1)^1 = -1,$$

$$W_2(5) = (-1)^{(0\cdot1+1\cdot0+0\cdot1)} = (-1)^0 = 1, \qquad W_7(5) = (-1)^{(1\cdot1+1\cdot0+1\cdot1)} = (-1)^2 = 1,$$

$$W_2(6) = (-1)^{(0\cdot0+1\cdot1+0\cdot1)} = (-1)^1 = -1, \qquad W_7(6) = (-1)^{(1\cdot0+1\cdot1+1\cdot1)} = (-1)^2 = 1,$$

$$W_2(7) = (-1)^{(0\cdot1+1\cdot1+0\cdot1)} = (-1)^1 = -1; \qquad W_7(7) = (-1)^{(1\cdot1+1\cdot1+1\cdot1)} = (-1)^3 = -1.$$

Проте для скорочення кількості операцій при побудові системи дискретних функцій Уолша використовується рекурентний матричний спосіб [49, 50]:

$$\begin{aligned} H_1 &= [1], \\ H_2 &= \begin{bmatrix} H_1 & H_1 \\ H_1 & -H_1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \\ &\vdots \\ H_{2^i} &= \begin{bmatrix} H_{2^{i-1}} & H_{2^{i-1}} \\ H_{2^{i-1}} & -H_{2^{i-1}} \end{bmatrix}, \end{aligned} \tag{2.4}$$

де i – рівень сигналу системи функцій Уолша. Матриці H_{2^i} називають матрицями Адамара. В отриманій матриці кожен рядок відповідає значенню функції ДФУ в точках відліку.

Наведемо приклад отримання системи дискретних функцій Уолша при довжині сигналу $N = 2^3 = 8$ за допомогою використання рекурентного співвідношення (2.4).

$$H_3 = \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}.$$

Графік отриманих дискретних функцій наведено на рис. 2.3.

Очевидно, що результати, отримані за допомогою формул (2.3) і (2.4) аналогічні, проте для випадку використання рекурентної матричної формули (2.4) кількість розрахунків значно скорочується.

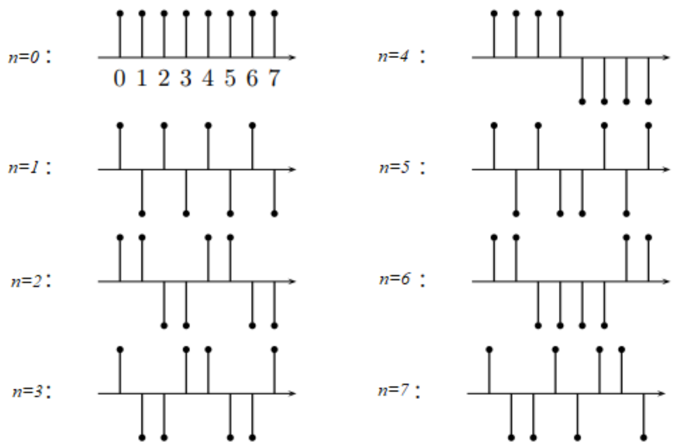


Рис. 2.3 Графіки дискретних функцій Уолша при довжині сигналу $N = 2^3 = 8$.

Дискретні функції Уолша володіють тими же властивостями, що й неперервні функції, на основі яких дані функції базуються [49-51]:

1. ДФУ ортонормовані:

$$\sum_{k=0}^{N-1} wal_i\left(k\right) wal_j(k) = \begin{cases} 1 & при \quad i = j, \\ 0 & при \quad i \neq j, \end{cases}$$

де k – номер відліку функції

2. Математичне очікування дискретних функцій Уолша нульове:

$$\sum_{k=0}^{N-1} wal_i\left(k\right) = 0, \quad i = 1, 2, ..., N-1.$$

3. Властивість мультиплікативності:

$$wal_i(k) wal_j(k) = wal_p(k),$$

де $p = i \oplus j$.

4. Функція симетричним відносно середини інтервалу ($x = 0,5$) відповідають парні значення номеру функцій n і навпаки.

5. Система функцій Уолша має область визначення $\left[0,1\right]$. Поза даним інтервалом функції періодично повторюються.

2.2. Лінійні функції на базі функцій Уолша

Для отримання лінійних ортонормованих функцій, які визначатимуть траєкторію точки в проміжних точках випадкового блукання, доцільно використовувати видозмінену систему функцій Уолша. Для цього застосуємо процедуру Грамма-Шмідта до системи функцій Уолша. Процедура Грамма-Шмідта дозволяє на основі лінійно-незалежносї системи функцій побудувати ортонормовану систему [51]:

$$\begin{aligned} \psi_i(x) &= \varphi_i(x) - \sum_{n=0}^{i-1} R_n(x) \int_0^1 \varphi_i(x) R_n(x) dx, \\ R_i(x) &= \frac{\psi_i(x)}{\sqrt{\int_0^1 \psi_i^2(x) dx}}, \end{aligned} \tag{2.5}$$

де $\varphi_0(x),\varphi_1(x),...,\varphi_n(x)$ – початкова система функцій, $R_1(x),R_2(x),...,R_n(x)$ – система ортонормованих функцій.

Перевіримо систему лінійних функцій $\varphi_0(x),\varphi_1(x),\varphi_2(x),\varphi_3(x)$ (рис. 2.3) на лінійну залежність за допомогою визначника Грама [51]:

$$\Gamma = \begin{bmatrix} \langle\varphi_0,\varphi_0\rangle & \langle\varphi_0,\varphi_1\rangle & \langle\varphi_0,\varphi_2\rangle & \langle\varphi_0,\varphi_3\rangle \\ \langle\varphi_1,\varphi_0\rangle & \langle\varphi_1,\varphi_1\rangle & \langle\varphi_1,\varphi_2\rangle & \langle\varphi_1,\varphi_3\rangle \\ \langle\varphi_2,\varphi_0\rangle & \langle\varphi_2,\varphi_1\rangle & \langle\varphi_2,\varphi_2\rangle & \langle\varphi_2,\varphi_3\rangle \\ \langle\varphi_3,\varphi_0\rangle & \langle\varphi_3,\varphi_1\rangle & \langle\varphi_3,\varphi_2\rangle & \langle\varphi_3,\varphi_3\rangle \end{bmatrix}, \tag{2.6}$$

$$\langle\varphi_i,\varphi_j\rangle = \int_a^b \varphi_i(x)\varphi_j(x)dt\,.$$

З урахуванням (2.2) матриця (2.6) набуде вигляду:

$$\Gamma = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Оскільки $\det(\Gamma)\neq 0$, то система функцій лінійно незалежна і процедура Грама-Шмідта може бути використана до вихідної системи функцій Уолша для отримання лінійних функцій.

Після застосування процедури ортогоналізації Грама-Шмідта лінійні функції на базі функцій Уолша описуються наступним виразом:

$$\begin{aligned} &W_{l0}(x) = 1 \\ &W_{\ln}(x) = wal_{nk}(n\sqrt{3} - k\sqrt{3}) \end{aligned}, \tag{2.7}$$

де n – номер функції; wal_{nk} – значення n-ої функції Уолша на k-му інтервалі розбиття, $\{n,k\} = 0...2^m - 1$.

Система даних лінійних функцій, отриманих за допомогою (2.7), при $N = 4$ зображена на рис. 2.4.

Система лінійних функцій на базі функцій Уолша має ті самі властивості, що й звичайні неперервні функції Уолша, на основі яких нові функції були отримані:

1. Функції Уолша ортонормовані:

$$\int_0^1 wal_{li}(x) wal_{lj}(x) dx = \begin{cases} 1 & при \quad i = j, \\ 0 & при \quad i \neq j. \end{cases}$$

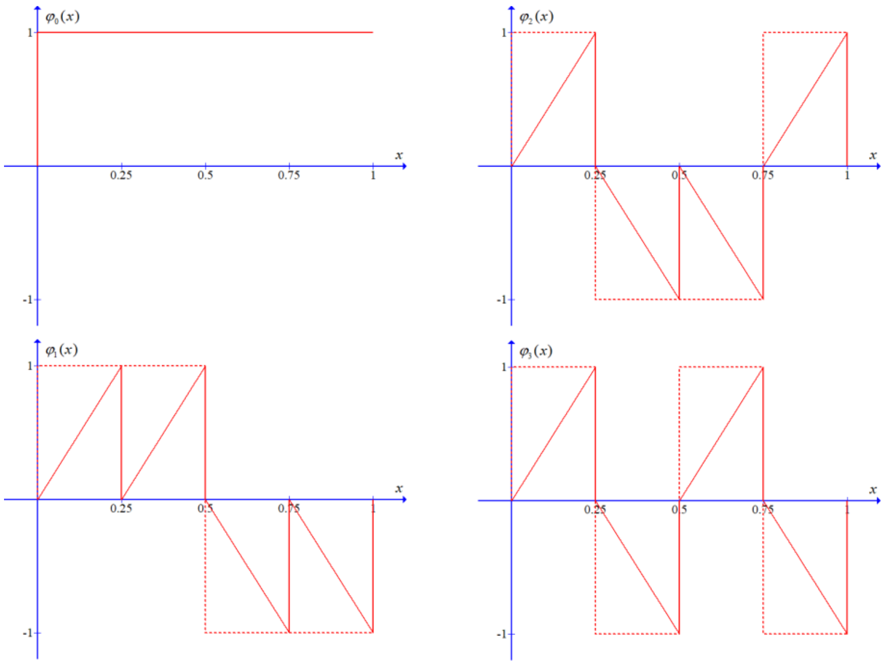


Рис. 2.4. Система лінійних функцій Уолша

2. Математичне очікування функцій Уолша для всіх $i \neq 0$:

$$\int_0^1 wal_{li}(x) dx = 0, \quad i = 1, 2, ..., N - 1.$$

3. Властивість мультиплікативності. Тобто добуток двох функцій Уолша є функцією Уолша тієї ж системи:

$$wal_{li}(x) wal_{lj}(x) = wal_{lk}(x),$$

де $k = i \oplus j$, $\oplus$ – порозрядне сума за модулем два.

4. Функція симетричним відносно середини інтервалу ($x = 0,5$) відповідають парні значення номеру функцій n і навпаки.

5. Система функцій Уолша має область визначення $[0,1]$. Поза даним інтервалом функції періодично повторюються.

2.3. Особливості побудови дискретних функцій Уолша

Як було сказано раніше, дискретні функції за рахунок відносної легкості побудови відіграють особливу роль при використанні на електронно-обчислювальних машинах. Тому для мінімізації обчислювального часу, затраченого для побудови системи лінійних функцій, доцільно створити систему дискретних лінійних функцій на базі функцій Уолша, які будуть також знаходитися за допомогою рекурентних співвідношень і матриць Адамара:

$$W_{dn}(k) = sign\left(W_{\ln}(k)\right), \tag{2.8}$$

де $sign$ – сигнум функція, $W_{\ln}(k)$ – значення лінійної функції Уолша під номером n на k-тому інтервалі розбиття.

Виходячи з виразу (2.8) дискретні функції Уолша матимуть вид звичайних ДФУ (рис. 2.3).

Вираз (2.8) підходить для побудови дискретних функцій, упорядкованих за будь-якою системою.

2.4. Математична модель випадкового блукання з використанням лінійних ортогональних функцій

В даній математичній моделі використовуються дискретні лінійні функції на базі функцій Уолша, що дозволить зменшити кількість обчислень при комп’ютерній реалізації даної моделі.

Як було наведено вище, випадкове блукання описується наступним виразом:

$$Y_n = Y_0 + \sum_{i=1}^{n} Y_i ,$$

$$Y_i = \left(Y_1, Y_1, \ldots, Y_k\right),$$

де Y_n – значення випадкового блукання на n-тому кроці (в n-й момент часу); Y_0 – початкове значення випадкового блукання, Y_i – випадковий процес розмірності k, який визначає значення кроку випадкового блукання на i-тому кроці (в i-й момент часу), причому $i < n$.

Для визначення траєкторії в проміжних точках використовується розглянуті вище дискретні функції. При чому кількість нових проміжних точок визначається кількістю дискретних відліків (розмірністю системи) N, кількість яких знаходиться за допомогою співвідношення:

$$N = 2^m,$$

де m – рівень сигналу, $m \in N$.

Використання даних функцій полягатиме в наступному. На кожному кроці для визначення проміжних точок спершу випадковим чином обиратимуться дискретні функції (окрім першої) wal_{kN}, кількість яких дорівнює розмірності випадкового процесу k, а розмірність системи N визначає кількість проміжних точок. Далі генеруються вектор випадкових значень R_{kN}, який також відповідає розмірності випадкового процесу k і системи лінійних функцій Уолша N. Після цього знаходиться добуток $A_{kN} = R_{kN} wal_{d\,kN}$, який сумується з випадковим значенням Y_i, яке отримується генератором випадкового блукання. Враховуючи вищесказане, випадкове блукання з використанням лінійних функцій на базі функцій Уолша описуватиметься виразом:

$$\begin{aligned} Y_n &= Y_0 + \sum_{i=1}^{n} R_{kN} wal_{kN} + Y_i, \\ Y_i &= \left(Y_0, Y_1, \ldots, Y_k\right), \end{aligned} \tag{2.9}$$

де Y_n – значення випадкового блукання на n-тому кроці (в n-й момент часу); Y_0 – початкове значення випадкового блукання; Y_i – випадковий процес розмірності, який визначає значення кроку випадкового блукання на i-тому кроці (в i-й момент часу), причому $i < n$; wal_{kN} – матриця, яка

складається з k дискретних функцій Уолша, кількість відліків яких N; R_{kN} – матриця випадкових значень з розмірністю $[k, N]$.

Розглянемо приклад побудови випадкового блукання з використанням формули (2.9) лінійних функцій Уолша при $N=4$, $k=2$ і початковим нульовим значенням.

Нехай на деякому кроці випадкового блукання була отримана наступна матриця випадкових значень $R_{2,4}$ і обрані дві функції (оскільки $k=2$) дискретні функції Уолша, які утворюють матрицю wal_{kN}:

$$R_2 = \begin{bmatrix} r_{12} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \end{bmatrix},$$

$$wal_{kN} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \end{bmatrix}.$$

Тоді їх добутком називатимемо:

$$A_{kN} = R_{kN} wal_{kN} = \sum_{i=1}^{k} \sum_{j=1}^{N} w_{ij} r_{ij} = \sum_{i=1}^{k} \sum_{j=1}^{N} a_{ij} .$$

Що еквівалентно матриці:

$$A_{kN} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \end{bmatrix}.$$

Після цього до отриманого добутку A_{kN} додається значення випадкового процесу на даному кроці наступним чином:

$$A_{kN} + Y_i = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & Y_1 \\ a_{21} & a_{22} & a_{23} & a_{24} & Y_2 \end{bmatrix}.$$

Аналізуючи отримані формули, встановлюються наступні властивості нової моделі:

1. Початковий закон розподілу кроку випадкового блукання не змінюється.

2. Кількість проміжних точок дорівнює розмірності системи дискретних функцій а базі функцій Уолша.

3. Проміжні точки не змінюють початкову траєкторію.

Висновки до другого розділу

В даному розділі були розглянуті неперервні і дискретні функції Уолша, а також їх основні властивості. На основі процедури Грама-Шмідта було отримано аналітичні вирази для лінійних функцій на базі функцій Уолша, а також їх дискретного представлення. Нові системи функцій володіють всіма властивостями функцій Уолша, на основі яких вони були побудовані.

Також була розроблена математична модель випадкового блукання з використанням лінійних функцій на базі функцій Уолша. Було детально розглянуто приклад побудови даної моделі, а також наведені властивості даної моделі.

3. ПРОГРАМНА РЕАЛІЗАЦІЯ ВИПАДКОВОГО БЛУКАННЯ З ВИКОРИСТАННЯМ ОРТОГОНАЛЬНИХ РЯДІВ

3.1. Короткий опис мови програмування

Python – це одна з найпопулярніших високорівневих мов програмування загального призначення, створена Гідо ван Россумом і вперше випущена у 1991 році. Ця скриптова мова дозволяє писати системні скрипти та програми максимально швидко, при цьому пропонуючи лише незначне падіння у швидкості виконання програми у порівнянні з мовою програмування C. Для забезпечення достатньо швидкого виконання використовується правило EAFP (Easier to Ask for Forgiveness than Permission). Це правило визначає стиль кодування, заснований на припущенні того, що код працює правильно, що ключі є в словниках, і на використанні try / except та пійманні виключень, якщо це не так. Це відрізняє Python від багатьох поширених мов, у тому числі і C.

Python має динамічну типізацію та автоматичне управління пам’яттю і підтримує декілька парадигм програмування, включаючи об'єктно-орієнтоване, імперативне, функціональне програмування, і процедурні стилі.

На даний момент активно використовуються дві основні версії Python:

1. Python 3.x - поточна версією, яка знаходиться у стадії активного розвитку.

2. Python 2.x – попередня версія, яка вважається застарілою та отримуватиме лише оновлення безпеки до 2020 року, при цьому новий функціонал додаватися не буде. Багато проектів все ще використовують Python 2, хоча перехід на Python 3 не є складним.

Стандартна бібліотека Python надає доступ до інструментів для вирішення різного роду завдань, включаючи та не обмежуючись математичними функціями, роботи з файлами, регулярними виразами,

мережею, інтернет сайтами та іншим. Весь функціонал стандартної бібліотеки складається з великої кількості модулів – файлів, що містять оголошення та визначення функцій, що реалізовують функціонал стандартної бібліотеки. Бібліотека містить наступні компоненти:

1. Типи даних – об’єкти, що можуть зберігати конкретний набір значень; їх зазвичай вважають частиною «ядра» мови, наприклад, числа та списки. Для стандартних типів ядро мови Python визначає форму літералів і накладає деякі обмеження на їх семантику, але повністю не визначає її.
2. Функції – набір інструкцій для реалізації логіки обробки даних;
3. Виключення для обробки помилок.

Деякі модулі написані на мові C і вбудовані в інтерпретатор Python, інші написані на Python та імпортовані у вигляді вихідного коду. Деякі модулі забезпечують інтерфейси, які є специфічними виключно для Python, наприклад, виводить класичні стеки; деякі пропонують інтерфейси, які є специфічними для конкретних операційних систем, такі як доступ до конкретного обладнання; інші надають інтерфейси, які є спеціальними для конкретної області, запропонованої, як, наприклад, Internet. Деякі модулі доступні у всіх версіях і порталі Python; інші доступні тільки тоді, коли операційна система підтримує або вимагає їх; інші доступні тільки тоді, коли був вибраний конкретний параметр конфігурації при компіляції та встановленні Python.

При створенні програмного забезпечення були використані Jupyter Notebook та Google Colab:

1. Jupyter Notebook – неймовірно потужний інструмент для інтерактивної розробки та презентації проектів в області наук про дані. Блокнот (notebook) об'єднує код та його вивід у єдиний документ, який у свою чергу об'єднує візуалізацію, текст, математичні рівняння та інше. На даний момент Jupyter Notebook є стандартним середовищем для науковців, які використовують дистрибутив Anaconda

2. Google Colab – це безкоштовний хмарний сервіс на основі Jupyter Notebook. Google Colab надає все необхідне для наукових розрахунків та машинного навчання прямо в браузері, а також дозволяє використовувати GPU для розрахунків.

При створенні програмного забезпечення були використані наступні бібліотеки та модулі, їх складові:

1) time – модуль зі стандартної бібліотеки мови програмування Python містить масу корисних методів для роботи з часом. З його допомогою можна отримувати інформацію про поточну дату і час з точністю до мілісекунд, виводити ці відомості у необхідному форматі, а також керувати ходом виконання програми, додаючи затримки по таймеру.

2) numpy - це бібліотека мови Python, що додає підтримку великих багатовимірних масивів і матриць, разом з великою бібліотекою високорівневих (і дуже швидких) математичних функцій для операцій з цими масивами. Бібліотека numpy надає функціонал, який можна порівняти з середовищем Matlab.

3) numpy.random – модуль random бібліотеки numpy дозволяє створювати псевдовипадкові числа з потрібним розподілом, серед яких використовуються наступні:

3.1) numpy.random.normal(avg, std, size) – відобразити псевдовипадкову вибірку з нормальним (гаусівським) законом розподілу з середнім значенням avg, середньоквадратичним відхиленням std та розміром вибірки size.

3.2) numpy.random.exponential(scale, size) – відобразити псевдовипадкову вибірку з експоненційним законом розподілу з коефіцієнтом інтенсивності scale та розміром вибірки size.

3.3) numpy.random.gamma(shape, scale, size) – відобразити псевдовипадкову вибірку з законом розподілу гамма з параметрами розподілу shape та scale та розміром вибірки size.

3.4) numpy.array(object, dtype, copy, order, subok, ndmin) – функція, що створює масив типу ndarray зі списку чи іншого стандартного типу, де object – масив або об’єкт, що має метод __array__ для його приведення до масиву, dtype – тип, що буде використовуватися для зберігання значень у масиві, copy визначає, чи потрібно копіювати масив, order визначає формат розташування значень масиву у пам’яті (за рядками як у мові програмування C, чи за стовпцями як у Fortran), subok визначає чи потрібно копіювати наслідувані класи з object, чи потрібно їх приводити до базового класу, ndmin – мінімальна розмірність матриці, на основі якої має бути сформований масив.

3.5) numpy.ones(shape, dtype, order) – функція, що повертає новий масив, заповнений 1, shape визначає розмірність матриці в основі масиву, dtype та order мають таке ж призначення, що і в numpy.array.

4) operator – модуль, що експортує набір ефективних функцій, що відповідають внутрішнім операторам Python. Наприклад, operator.add (x, y) еквівалентний виразу x + y.

4.1) operator.ne(x, y) - функція, еквівалент x <= y.

4.2) operator.gt(x, y) - функція, еквівалент x > y.

4.3) operator.ge(x, y) - функція, еквівалент x >= y.

5) plotly – це інтерактивна бібліотека для створення інтерактивних графіків з відкритим вихідним кодом, яка підтримує понад 40 унікальних типів діаграм та графіків, що охоплюють широкий спектр статистичних, фінансових, географічних, наукових та тривимірних випадків використання. Побудований поверх бібліотеки Plotly JavaScript (plotly.js), plotly.py дозволяє користувачам Python створювати прекрасні інтерактивні візуалізації на основі веб-сторінок, які можна відображати в ноутбуках Jupyter, зберігати у самостійних HTML-файлах або ж відображувати в браузері.

5.1) plotly.graph_objects – модуль для роботи з конкретними типа графіків та діаграм.

5.2) plotly.graph_objects.Figure() – створює об’єкт, що буде зберігати дані про вікно та графіки, що на ньому містяться.

5.3) plotly.graph_objects.Histogram(name, … x, y, …, xaxis, yaxis, …, nbinsx, nbinsy, …) – функція для створення гістограми з заданими даними та параметрами, де name – назва гістограми, що відображається над нею, x – відліки по осі x, y – дані для побудови гістограм, xaxis, yaxis – параметри для налаштування відображення осей, nbinsx – максимальна кількість стовбців гістограми по осі x, nbinsy – максимальна кількість стовбців гістограми по осі x .

5.4) plotly.graph_objects.Histogram2dContour(name, visible, … x, y, …, xaxis, yaxis, …, nbinsx, nbinsy, …, line, …, contours, …) – функція для створення двовимірної гістограми, де visible визначає, чи є графік видимим, line визначає параметри та вигляд ліній, contour визначає параметри та вигляд зафарбованих областей гістограми, інші параметри аналогічні до попередніх.

5.5) plotly.graph_objects.Scatter(name, …, mode, …, x, …, y, …, xaxis, yaxis, marker, …) – функція для створення діаграми розсіювання, де mode визначає вигляд точок на діаграмі – те чи вони будуть підписані, чи будуть виділені, marker – графічний вигляд точок – їх розмір, форма, інші параметри аналогічні до попередніх.

5.6) plotly.graph_objects.Scatter3d(name, …, mode, …, x, …, y, …, z, …, marker, …) – функція для створення тривимірної діаграми розсіювання, де z визначає масив з координатами z точок, інші параметри аналогічні до попередніх.

5.7) plotly.subplots – модуль для роботи з багато чисельними графіками на одному вікні.

5.8) plotly.subplot.make_subplots(rows, cols) – функція, що задає кількість графіків у вікні по горизонталі та вертикалі та повертає об’єкт даного вікна.

5.9) plotly.subplot.add_trace(figure) – функція, що додає графік figure до вікна.

5.10) plotly.subplot.update_layout(…) – функція, що оновлює параметри відображення усіх графіків водночас згідно з переданими параметрами.

5.11) plotly.plot(figure) – функція що створює веб-сторінку з графіками, заданими в figure, повертає URL-адресу на дану сторінку та намагається її відкрити.

3.2. Алгоритм роботи програмного забезпечення

Блок схема алгоритму роботи програми, яка моделює випадкове блукання з використанням лінійної функції на базі функцій Уолша наведено рис 3.1.

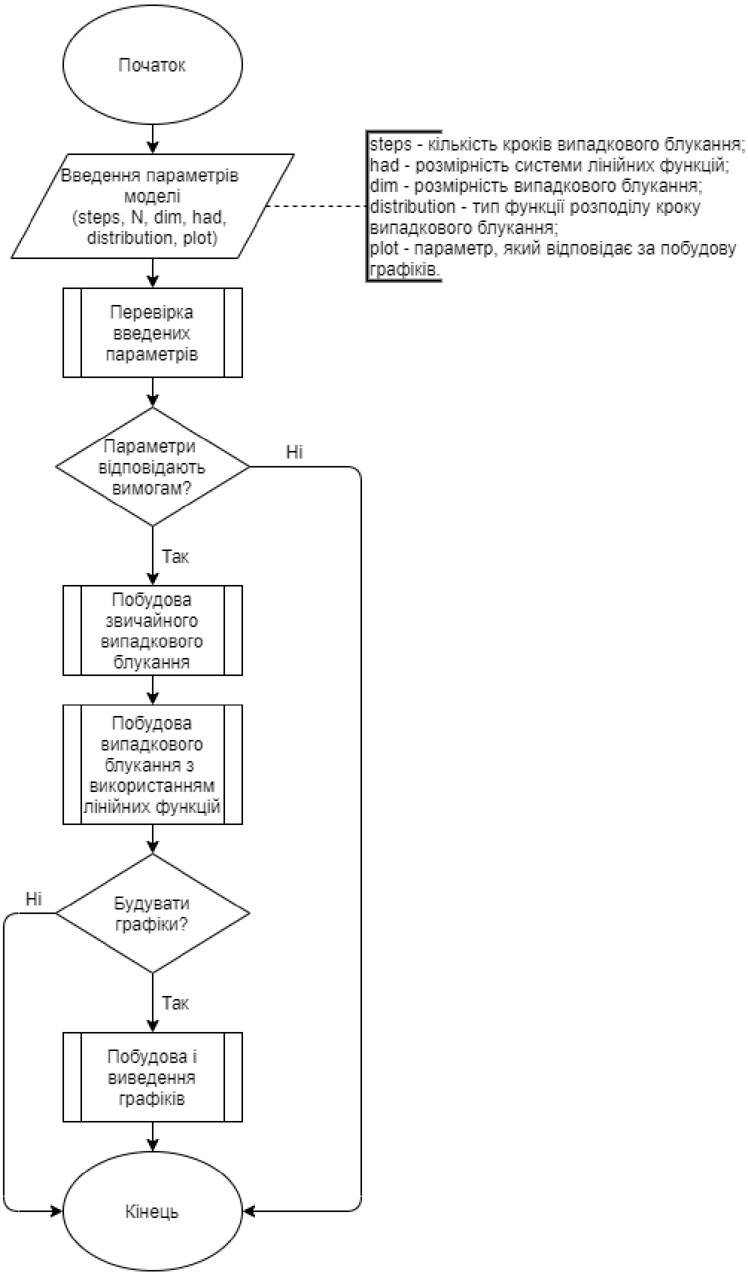


Рис. 3.1. Блок-схема алгоритму

3.3. Порівняння моделей

Основним параметром для порівняння буде тест швидкодії. Для цього використовуватиметься модуль timeit стандартної бібліотеки Python, який дозволяє вимірювати швидкість виконання переданого в якості аргументу коду.

При аналізі швидкодії тест проводився для різної кількості кроків випадкового блукання (10, 100, 300, 500, 1000) і різної розмірності блукання

(1, 2, 3) і при розмірності системи лінійних функцій $N = 4$. Результати аналізу наведені на рис. 3.2.

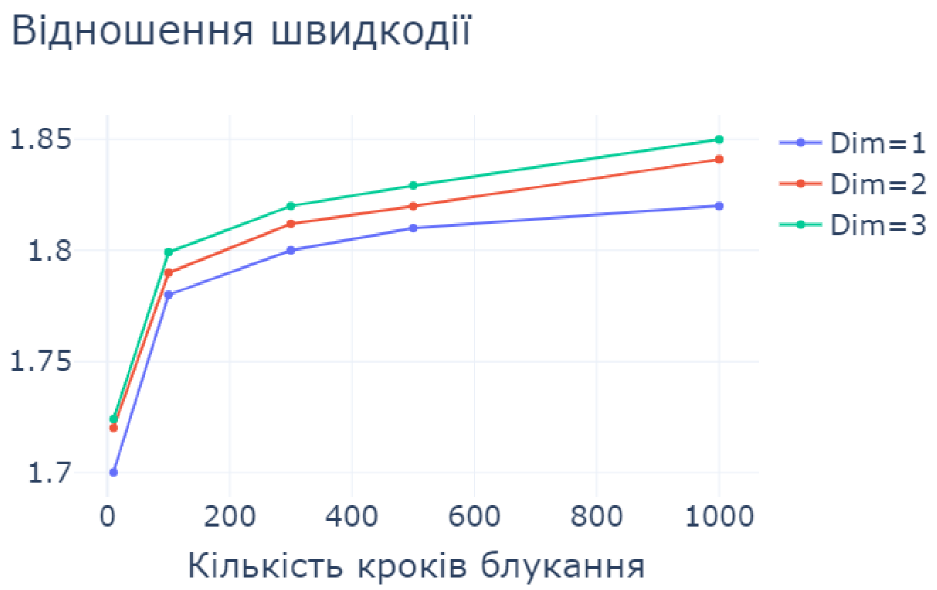


Рис. 3.2.Графік відношення швидкодії

В загальному випадку звичайне блукання має час виконання менший в середньому в 1.8 разів. Проте варто враховувати те, що при використанні генератора з лінійними функція при $N = 4$ кількість точок в 5 разів більша, ніж у звичайного генератора, що говорить про те, що генератор з використанням лінійних функцій працює швидше.

Висновки до третього розділу

В даному розділі були наведені особливості мови програмування Python. Дана мова характеризується високою швидкістю розробки програмного забезпечення і багатоплатформністю, що дозволяє використовувати скрипти написані на даній мові майже на всіх сучасних присторях. Недоліком даної мови являється відносно мала швидкість виконання програми у порівнянні з такими компільованими мовами

програмування як C, C++, Rust. Також було розглянуті основні модулі і пакети, які використовувалися для розробки програмного забезпечення.

Було наведено блок-схема алгоритму роботи генератора з використанням лінійних функцій Уолша, а також проведено порівняльний аналіз швидкодії генераторів. При однаковій кількості кроків випадкового блукання розроблений алгоритм являється дещо швидшим.

РОЗДІЛ 4. РОЗРОБКА СТАРТАП-ПРОЕКТУ

Стартап або стартап-компанія (від англ. Start-up - запускати) – це тип бізнесу, спрямований на отримання доходу шляхом реалізації принципово нової ідеї.

Початком роботи над будь-яким стартапом є створення його прототипу.

Далі прототип перетворюється в повноцінний продукт, трансформується і розвивається, багато разів масштабуючись. Протягом усього цього часу стартап кілька разів залучає інвестиції, команда стартапу зростає, а складність продукту збільшується.

В кінцевому рахунку, метою створення стартапу є його продаж великої корпорації або виведення його акцій на біржу і продовження роботи в якості окремої компанії.

Етапи розроблення стартап-проекту:

1. Маркетинговий аналіз стартап-проекту (опис ідеї проекту, вибір загального напрямку використання товару або послуги, аналіз ринкових можливостей)

2. Організація стартап-проекту (визначаються початкові витрати на запуск проекту та планові витрати)

3. Фінансово-економічний аналіз та оцінка ризиків проекту (визначається обсяг інвестиційних витрат та основні інвестиційні параметри проекту)

4. Заходи з комерціалізації проекту (визначення цільової групи інвесторів, планування маркетингової компанії)

4.1. Опис ідеї проекту

Опис ідеї стартап-проекту наведено в табл. 4.1, визначення сильних, слабких та нейтральних характеристик ідеї проекту в табл. 4.2.

Таблиця 4.1

Опис ідеї стартап - проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Створення і комп'ютерна реалізація математичної моделі випадкового блукання з використанням лінійних функцій.	1. Промислова електроніка. 2. Фізика. 3. Економіка. 4. Біологія.	Використання програмного забезпечення для аналізу різноманітних задач прикладної науки.

Таблиця 4.2

Визначення сильних, слабких та нейтральних характеристик ідеї проекту

Техніко-економічні характеристики ідеї	*(потенційні) товари/концепції конкурентів*				*W слабка сторона*	*N нейтральна сторона*	*S сильна сторона*
	Мій проект	*Конкурент1*	*Конкурент2*	*Конку-рент3*			
Можливість побудови траєкторії у проміжних точках	+	-	-	-			+
Швидкодія	+/-	+	-	+	+		
Багатоплатформність програмного забезпечення	+	-	-	+			+

4.2. Технологічний аудит ідеї проекту

Технологічна здійсненність ідеї проекту наведено в табл. 4.3.

Таблиця 4.3

Технологічна здійсненність ідеї проекту

Ідея проекту	*Технології її реалізації*	*Наявність технологій*	*Доступність технологій*
Створення і реалізація випадкового блукання з	Розробка, дослідження,	Наявна	Доступна

використанням лінійних функцій	програмування;		

4.3. Аналіз ринкових можливостей запуску стартап-проекту

Попередню характеристика потенційного ринку наведено в табл. 4.4. В табл. 4.5 наведено характеристика потенційних клієнтів.

Таблиця 4.4.

Попередня характеристика потенційного ринку стартап проекту

Показники стану ринку (найменування)	*Характеристика*
Кількість головних гравців, од	4
Загальний обсяг продаж, грн/ум.од	-
Динаміка ринку (якісна оцінка)	Зростає
Наявність обмежень для входу (вказати характер обмежень)	Масове розповсюдження програмних продуктів
Специфічні вимоги до стандартизації та сертифікації	Згідно ДСТУ
Середня норма рентабельності в галузі (або по ринку), %	13%

Таблиця 4.5

Характеристика потенційних клієнтів стартап-проекту

Потреба, що формує ринок	*Цільова аудиторія (цільові сегменти ринку)*	*Відмінності у поведінці різних потенційних цільових груп клієнтів*	*Вимоги споживачів до товару*
Потреба в якісній математичній моделі і програмному забезпеченні для потреб моделювання в галузі прикладної науки	Дослідницькі лабораторії, стартапи	Експлуатація згідно сертифікаційних правил	- до продукції: Ефективність Надійність Наявність зворотного зв’язку - до компанії-постачальника: Професіоналізм Чесність Порядність Технічна підтримка

В табл. 4.6 наведено фактори загроз, їх зміст та можлива реакція компанії.

Таблиця 4.6

Фактори загроз

Фактор	*Зміст загрози*	*Можлива реакція компанії*
Збій у роботі програмного забезпечення	Неправильне використання програмного забезпечення	Оновлення програмного забезпечення
Невисока швидкодія	Низька швидкодія у порівнянні з компільованими рішеннями	Перехід на компільовану мову програмування

В табл. 4.7 наведено фактори можливостей, їх зміст та можливу реакцію компанії.

Таблиця 4.7

Фактори можливостей

Фактор	*Зміст можливості*	*Можлива реакція компанії*
Якість моделювання випадкового блукання	Якісна модель випадкового блукання	Залучення нових клієнтів за допомогою маркетингу

В табл. 4.8 наведено результати аналізу конкуренції на ринку, а в табл. 4.9 наведено обґрунтування факторів конкурентоспроможності.

Таблиця 4.8

Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	*В чому проявляється дана характеристика*	*Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)*
1. Тип конкуренції: чиста	В кого краще – в того	Покращення якості

	купують	програмного забезпечення і обслуговування
2. За рівнем конкурентної боротьби: світова	Належить до вузького ринку збуту	Розширення функціоналу та орієнтації на користувачів
3. За галузевою ознакою: міжгалузева	Може бути використана в декількох галузях, які в своїй роботі використовують модель випадкового блукання	Розширення функціоналу та галузей застосування
4. Конкуренція за видами товарів: товарно-видова	Відрізняється здатністю побудови траєкторії в проміжних точках	Розширення функціональних здатностей програмного забезпечення
5. За характером конкурентних переваг: цінова та нецінова	Чим дешевше – тим привабливіше. Чим краще – тим рентабельніше	Покращення цінової політики та якості товару
6. За інтенсивністю: не марочна	Не жорстка конкуренція	Агресивні форми піару

Таблиця 4.9

Обґрунтування факторів конкурентоспроможності

Фактор конкурентоспроможності	*Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)*
Ціна	Серед схожих по характеристикам систем обиратимуть ту, яка дешевше
Якість	Серед схожих по ціні систем обиратимуть яка має кращі характеристики
Відомість	При рівності двох перших факторів обиратимуть більш відомий товар

Результати порівняльного аналізу сильних та слабких сторін стартап-проекту наведено в табл. 4.10

Таблиця 4.10

Порівняльний аналіз сильних та слабких сторін «RandWalk»

Фактор конкурентоспроможності	*Бали 1-20*	*Рейтинг товарів-конкурентів у порівнянні з* DP						
		–3	*–2*	*–1*	*0*	*+1*	*+2*	*+3*
Ціна						+		
Якість						+		

Відомість				+				

В табл. 4.11 наведено аналіз слабких, сильних сторін стартап-проекту, а також можливості та загрози.

Таблиця 4.11

SWOT- аналіз стартап-проекту

Сильні сторони: Оптимізоване програмне забезпечення	Слабкі сторони: Мала швидкодія
Можливості: Можливість побудови випадкового блукання в проміжних точках	Загрози: Поява більш ефективного програмного забезпечення

4.4. Розробка маркетингової програми стартап-проекту

Для розроблення маркетингової програми стартап-проекту перш за все необхідно визначити базову стратегію конкурентної поведінки. Результати наведені в табл. 4.12

Таблиця 4.12

Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку?	*Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?*	*Чи буде компанія копіювати основні характеристики товару конкурента, і які?*	*Стратегія конкурентної поведінки**
Ні	Так	Так	Стратегія заняття конкурентної ніші

В табл. 4.13 наведено визначення ключових переваг концепції потенційного товару, тобто, вигоду, яку пропонує дана система для споживачів.

Таблиця 4.13

Визначення ключових переваг концепції потенційного товару

Потреба	*Вигода, яку пропонує товар*	*Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)*
Підвищення ефективності використання енергії в системах з використанням ВДЕ	Якісне програмне забезпечення	Швидкодія на рівні компільованих мов програмування

В табл. 4.14 наведено рівень цін на товари замінники та аналоги, а також верхня та нижня межа ціни на аналогічне програмне забезпечення.

Таблиця 4.14

Визначення меж встановлення ціни

Рівень цін на товари-замінники	*Рівень цін на товари-аналоги*	*Рівень доходів цільової групи споживачів*	*Верхня та нижня межі встановлення ціни на товар/послугу*
100-110% від ціни нашого продукту	100-120% від ціни нашого продукту	10000 грн	3000/4000грн

Формування системи збуту наведено в табл. 4.15

Таблиця 4.15

Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	*Функції збуту, які має виконувати постачальник товару*	*Глибина каналу збуту*	*Оптимальна система збуту*
Роздрібна та оптова закупівля продукту	Збут та обслуговування (підтримка) програмного	Усі можливі канали збуту	Власна

	забезпечення	(глибока)	

Висновки до четвертого розділу

В результаті проведеного аналізу ринку, маркетингових досліджень та аналізу наявних науково-технічних рішень, можна зробити висновок щодо успішності розробленого стартап-проєкту. Отримана під час аналізу інфографіка свідчить про стабільний ріст ринку і збільшення попиту. Перешкодами для реалізації проекту і виходу на ринок є велика конкуренція на ринку, а також вузька клієнтська база. Що вимагає підвищення якості товару і обслуговування для конкурентоздатності.

Проте при правильному підході і агресивній маркетинговій кампанії проект має високі перспективи вдалого виходу на ринок і шанс на отримання прибутку з проекту. Тому подальша розробка і вихід на ринок є доцільними та рентабельними.

ВИСНОВКИ

У рамках виконання даної роботи було вирішено актуальну проблему створення математичної моделі випадкового блукання з використанням ортогональних рядів для потреб моделювання випадкової складової відновлюваних джерел енергії.

1. Було побудовано систему лінійних функцій на базі функцій Уолша, які використовуються для визначення проміжних точок випадкового блукання.

2. На основі лінійних функцій на базі функцій Уолша була створена нова математична модель випадкового блукання, а також її компютерна реалізація на мові програмування Python.

3. Було проведено порівняльний аналіз швидкодії отриманої компютерної програми з наявними рішеннями. Результати порівняння свідчать про незначну втрату швидкодії. Проте при цьому збільшується загальна кількість точок траєкторії випадково блукання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мартинюк В.І. «Покращення системи функцій Уолша». Збірник праць XI міжнародної науково-технічної конференції молодих вчених «Електроніка-2018», с. 284-288.
2. В.І. Мартинюк, К. С. Клен Лінійні функції на базі функцій Уолша, Мікросистеми, Електроніка та Акустика. – 2019. – Том 24, №1. – С. 24-39
3. Klenke, Achim. Probability Theory. A Comprehensive Course / Klenke, Achim., 2014.
4. Кремер Н.Ш. Теория Вероятностей и математическая статистика: Учебник для вузов – 2-е изд., перераб и доп. – М., 2004, – 573 с.
5. Лебедев А.В. Теория вероятностей и математическая статистика / Лебедев А.В, Фадеева Л.Н.. – Москва: Chapman and Hall, 2018. – 480 с.
6. Гмурман В.Е. Теория вероятностей и математическая статистика (4-е изд.). М.: Высшая школа, 1972.
7. Савельев Л.Я. Элементарная теория вероятностей. Часть 1. Новосибирск: НГУ, 2005.
8. Феллер В., Введение в теорию вероятностей и ее приложения. В 2-х томах. Т. 1.-М.: Мир, 1984.-528 с., ил. Г. Кантор. Труды по теории множеств. — М.: Наука, 1985. — 430 с.
9. Савельев Л.Я. Элементарная теория вероятностей. Часть 2. Новосибирск: НГУ, 2005.
10. Бернштейн С.Н. Теория вероятностей. М.-Л.: ГИ, 1927.
11. Прохоров Ю. В., Розанов Ю. А. Теория вероятностей (Основные понятия. Предельные теоремы. Случайные процессы) — М.: Главная редакция физико-математической литературы изд-ва «Наука», 1973. - 496 стр
12. Биллингсли П. Сходимость вероятностных мер. М.: Наука, 1977.

13. Бокс Дж. Дженкинс Г. Анализ временных рядов: прогноз и управление. Выпуск 1. М.: Мир, 1974.

14. Борель Э. Вероятность и достоверность. М.: Наука, 1969. – 112 с.

15. Венцтель Е.С. Теория вероятностей (4-е изд.). М.: Наука, 1969. – 576 с.

16. Боровков А. А. Теория вероятностей. – 2-е изд., перераб. и доп.- М.: Наука. Гл. ред. физ.-мат. лит. 1986.-432 с.

17. Гнеденко Б.В., Хинчин А.Я. Элементарное введение в теорию вероятностей (7-е изд.). М.: Наука, 1970.

18. Коралов Л.Б., Синай Я.Г. Теория вероятностей и случайные процессы – М.: МЦНМО, 2014. – 407 с.

19. Вентцель Е.С., Овчаров Л.А. Теория случайных процессов и ее инженерные приложения – 2-е изд. – М.: Высшая школа, 2000 – 383 с.

20. Sheldon Ross. Introduction to Probability Models / Sheldon Ross., 2019. – 842 с.

21. Sheldon Ross. A First Course in Probability / Sheldon Ross., 2018. – 552 с.

22. Jeffrey Seth Rosenthal. A First Look at Rigorous Probability Theory / Jeffrey Seth Rosenthal., 2006. – 219 с.

23. R. M. Dudley. Real Analysis and Probability / R. M. Dudley., 2002. – 555 с.

24. Колмогоров А. Н., Основные понятия теории вероятностей, 2 изд., М., 1974.

25. Ya G. Sinai. Probability Theory: An Introductory Course - Springer Textbook S. / Ya G. Sinai., 1992. – 148 с.

26. Миллер Б.М., Панков А.Р. Теория случайных процессов в примерах и задачах – М.: Физмат, 2002. – 320 с.

27. Прохоров С.А. Математическое описание и моделирование случайных процессов – Самара: Самар. гос. аэрокосм. ун-т, 2001. – 209 с.

28. Миллер Б.М., Панков А.Р. Теория случайных процессов в примерах и задачах – М.: Физматлит, 2002. – 320 с.

29. Кельберт М. Я., Сухов Ю. М., Вероятность и статистика в примерах и задачах. Т. 2: Марковские цепи как отправная точка теории случайных процессов и их приложения. – М.: МЦНМО, 2010.

30. Волков И.К., Зуев С.М., Цветкова Г.М. Случайные процессы – М.: Изд-во МГУ им. Н.Э. Баумана, 1999. – 448 с.

31. Вентцель А.Д. Курс теории случайных процессов – М.: Наука. Физматлит, 1996. – 400 с.

32. Кемени Дж., Снелл Дж. Конечные цепи Маркова – М.: Наука, 1970. – 272 с.

33. Тихонов В.И., Миронов М.А. Марковские процессы – М.: Советское радио, 1996 – 488 с.

34. Дорф Р., Бишоп Р. Современные системы управления – М.: Лаборатория базовых знаний, 2002. – 832 с.

35. Bartolucci F., Farcomeni A., Pennoni F. Latent Markov Models for Longtitudal Data -- New York: Chapman and Hall/CRC, 2012. – 252 p.

36. Кац М., Статистическая независимость в теории вероятностей, анализе и теории чисел, пер. с англ., М., 1963.

37. Чебышев П. Л., О средних величинах, в кн.: Полн. Собр. Соч., т. 2, М.- Л., 1947.

38. Тихонов В.И., Харисов В.Н. Статистический анализ и синтез радиотехнических устройств и систем. — Учебное пособие для ВУЗов. — М.: Радио и связь, 1991. — 608 с. — ISBN 5-256-00789-0.

39. А. В. Булинский, А. Н. Ширяев. Теория случайных процессов. Физматлит, 2005.

40. Баруча-Рид А. Т. Элементы теории марковских процессов и их приложения. — М. : Наука, 1969. — 512 с.

41. Гардинер К. В. Стохастические методы в естественных науках. — М. : Мир, 1986. — 528 с.

42. Скопенков, М.; Смыкалов, В.; Устинов, А. Случайные блуждания и электрические цепи – Матем. просв., сер. 3. — 2012. — Т. 16.

Шипилова, К.Ф.. Применение вейвлет-преобразований в нефтепромысловой практике / К.Ф. Шипилова, А.В. Байгушев // Автоматизация, телемеханизация и связь в нефтяной промышленности. – Москва, 2011. – № 7. – С. 45- 61.

43. Залманзон, Л.А. Преобразования Фурье, Уолша, Хаара и их применение в управлении, связи и других областях / Л.А. Залманзон. – М.: Наука, 1989. – 496 с.

44. Ахмед, Н. Ортогональные преобразования при обработке цифровых сигналов / Н. Ахмед, К.Р. Рао. – М.: Связь, 1980. – 248 с. 8.

45. Дагман, Э.Е. Быстрые дискретные ортогональные преобразования / Э.Е. Дагман, Г.А. Кухарев. – Новосибирск: Наука, 1983. – 230 с. 9.

46. Система функций Радемахера И Уолша в обработке сигналов – Гомель, 2011. – 3 с.

47. М. С. Беспалов В. А. Скляренко Функции Уолша и их приложения Учебное пособие.

48. Гоноровский И.С. Радиотехнические цепи и сигналы: Учебник для вузов – 4-е изд., перераб. и доп. – М.: Радио и связь, 1986. – 512 с.: ил.

49. Исмагилов, И.И. Об одном подходе к упорядочению систем дискретных функций Уолша / И.И. Исмагилов // Известия вузов. Радиоэлектроника. – 2006. – Т. 49, N 1. – С. 65–72.

50. Трахтман А.М. Основы теории дискретных сигналов на конечных интервалах / Трахтман А.М, Трахтман В.А.. – Москва: Советское радио, 1975. – 208 с.

51. Philip Franklin - A set of continuous orthogonal functions [Електронний ресурс]. – 1927. – Режим доступу до ресурсу: https://link.springer.com/article/10.1007%2FBF01448860.

ДОДАТКИ А. ЛІСТИНГ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

```
import time
import numpy as np
import operator as op
import numpy.random as rd
import plotly.offline as ply
import plotly.graph_objects as go
from scipy.linalg import hadamard
from plotly.subplots import make_subplots


def ne(arg):
    return op.ne(arg, 0)


def gt(arg):
    return op.gt(arg, 0)


def ge(arg):
    return op.ge(arg, 0)


def ok(arg):
    return True


class BaseError(Exception):
    pass


class WrongArgType(BaseError):
    pass


class WrongValue(BaseError):
    pass


class WrongArgsNumber(BaseError):
    pass


class Random_walk:
    DISTRIBUTIONS = {'normal': (rd.normal, (ok, ge)),
```

```
                'gauss': (rd.normal, (ok, ge)),
                'exp': (rd.exponential, ne),
                'exponential': (rd.exponential, ne),
                'gamma': (rd.gamma, (gt, gt))
                }

    def __init__(self, *args, steps=10000, dim=1, had=4, ax_distance=2, distribution='normal'):
        if distribution in Random_walk.DISTRIBUTIONS:
            self.distribution = distribution
            self.distribution_func, self.conditions = Random_walk.DISTRIBUTIONS[distribution]
        if Random_walk.validate_type(steps, dim):
            self.steps = steps
            self.dim = dim
        if Random_walk.validate_values(*args, conditions=self.conditions) and \
                Random_walk.validate_type(*args, types=(int, float)):
            self.args = args
            self.mean = args[0]
            self.std = args[1]
        self.had = had
        self.ax_distance = ax_distance
        self.had_matrix = hadamard(had)
        self.random = np.array(self.gen_random_num())
        self.path = np.array(self.gen_path())
        self.path_had, self.random_had = np.array(self.gen_path_had())

    @staticmethod
    def validate_type(*args, types=(int,)):
        for arg in args:
            if not isinstance(arg, types):
                raise WrongArgType(arg)
            return True

    @staticmethod
    def validate_values(*args, conditions):
        if len(args) > len(conditions):
            raise WrongArgsNumber('Too many args')
        elif len(args) < len(conditions):
            raise WrongArgsNumber('Not enough args')
        for index, cond in enumerate(conditions):
            if not cond(args[index]):
                raise WrongValue(args[index])
        return True

    def gen_random_num(self):
        res = []
        for index in range(self.dim):
            res.append(self.distribution_func(*self.args, self.steps))
        return res
```

```
def gen_random_num_rd(self):
    res = []
    for index in range(self.dim):
        res.append([self.distribution_func(*self.args) for i in range(self.steps)])
    return res

def gen_path(self):
    res = [[0] for i in range(self.dim)]
    for i in range(self.dim):
        for j in range(self.steps):
            res[i].append(res[i][-1] + self.random[i][j])
    return res

def gen_path_had(self):
    res = []
    rand = []
    for item in self.path:
        buff = [0]
        rd_buff = []
        for index in range(self.steps):
            rand_buff = np.random.normal(self.mean, self.std, self.had)
            buff.extend((np.linspace(item[index], item[index + 1], self.had) +
                         abs(rand_buff) * 0.1 * self.had_matrix[rd.randint(1, self.had - 1)]).tolist())
            buff.append(item[index + 1])
            rd_buff.append(rand_buff[0])
        res.append(buff)
        rand.append(rd_buff)
    return res, rand

def plot(self):
    if self.dim == 1:
        self.plot1d_path()
        self.plot1d_dist()
    elif self.dim == 2:
        self.plot2d_path()
        time.sleep(1)
        self.plot2d_dist()
    elif self.dim == 3:
        self.plot3d_path()
        time.sleep(1)
        self.plot3d_dist()

def plot1d_path(self):
    fig = go.Figure()
    x = [i * (self.had + 1) for i in range(self.steps + 1)]

    fig.add_trace(go.Scatter(y=self.path_had[0],
                             mode='lines+markers',
                             marker=dict(size=2, color='red'),
```

```
                         line=dict(width=1),
                         name='1D random walk with Walsh functions',
                         )
              )

    fig.add_trace(go.Scatter(x=x,
                         y=self.path[0],
                         mode='lines+markers',
                         marker=dict(size=1, color='blue'),
                         line=dict(width=1),
                         name='1D regular random walk',
                         )
              )

    fig.add_trace(go.Scatter(x=[0],
                         y=[self.path[0][0]],
                         mode='markers',
                         marker=dict(size=7, color='green'),
                         name='Start',
                         )
              )

    fig.add_trace(go.Scatter(x=[len(self.path_had[0])],
                         y=[self.path[0][-1]],
                         mode='markers',
                         marker=dict(size=7, color='black'),
                         name='End',
                         )
              )

    fig.update_layout(template="plotly_white",
                  autosize=True,
                  xaxis=dict(title='Steps'),
                  yaxis=dict(zeroline=False, showgrid=True),
                  title='Y',
                  font=dict(size=20),
                  legend=dict(font=dict(size=20),
                       bgcolor="White",
                       itemsizing='constant'
                       ),
                  )

    ply.plot(go.Figure(fig))

def plot1d_dist(self):
    nbins = int(self.steps ** 0.5)
    subplots = make_subplots(rows=2, cols=1)

    subplots.add_trace(go.Histogram(x=self.random[0], nbinsx=nbins,
```

```
                        name='Step distribution of 1D regular random walk',
                        histnorm="probability"
                        ),
                row=1, col=1
                )

    subplots.add_trace(go.Histogram(x=self.random_had[0], nbinsx=nbins,
                        name='Step distribution of 1D random walk with Walsh functions',
                        histnorm="probability"
                        ),
                row=2, col=1
                )

    subplots.update_layout(template="plotly_white",
                      autosize=True,
                      #width=1000,
                      # #height=1000,
                      xaxis=dict(showgrid=True, title='Step values'),
                      yaxis=dict(showgrid=True),
                      title='Probability',
                      font=dict(size=20),
                      legend=dict(font=dict(size=20),
                            bgcolor="White",
                            itemsizing='constant'
                            ),
                      )

    ply.plot(go.Figure(subplots))

def plot2d_dist(self):
    nbins = int(self.steps ** 0.5)
    fig = go.Figure()

    fig.add_trace(go.Histogram2dContour(x=self.random[0], y=self.random[1],
                         xaxis='x', yaxis='y',
                         line=dict(width=1),
                         contours=dict(coloring='none',
                                showlabels=True,
                                labelfont=dict(size=18)
                                ),
                         visible='legendonly',
                         name='Contour lines'
                         )
             )

    fig.add_trace(go.Scatter(x=self.random[0], y=self.random[1],
                      xaxis='x', yaxis='y',
                      mode='markers', marker=dict(size=2),
                      name='X-Y distribution'
```

```
            )
        )

        fig.add_trace(go.Histogram(x=self.random[0], yaxis='y2', name='X distribution', nbinsx=nbins,
                hovertemplate='value:%{x:.2f}' +
                    '<br>probability:%{y:.2f}<br>',
                histnorm="probability"
                )
        )

        fig.add_trace(go.Histogram(y=self.random[1], xaxis='x2', name='Y distribution', nbinsy=nbins,
                hovertemplate='value:%{y:.2f}' +
                    '<br>probability:%{x:.2f}<br>',
                histnorm="probability"
                )
        )

        fig.add_trace(go.Histogram2dContour(x=self.random_had[0], y=self.random_had[1],
                    xaxis='x', yaxis='y',
                    line=dict(width=1),
                    contours=dict(coloring='none',
                        showlabels=True,
                        labelfont=dict(size=18)
                        ),
                    visible='legendonly',
                    name='Contour lines'
                    )
        )

        fig.add_trace(go.Scatter(x=self.random_had[0], y=self.random_had[1],
                xaxis='x', yaxis='y',
                mode='markers', marker=dict(size=2),
                name='X-Y distribution'
                )
        )

        fig.add_trace(go.Histogram(x=self.random_had[0], yaxis='y2', name='X distribution', nbinsx=nbins,
                hovertemplate='value:%{x:.2f}' +
                    '<br>probability:%{y:.2f}<br>',
                histnorm="probability"
                )
        )

        fig.add_trace(go.Histogram(y=self.random_had[1], xaxis='x2', name='Y distribution', nbinsy=nbins,
                hovertemplate='value:%{y:.2f}' +
                    '<br>probability:%{x:.2f}<br>',
                histnorm="probability"
                )
        )
```

```
    fig.update_layout(template="plotly_white",
                      autosize=True,
                      xaxis=dict(zeroline=False,
                                 domain=[0, 0.85],
                                 showgrid=True,
                                 range=self.axes_range[0],
                                 ),
                      yaxis=dict(zeroline=False,
                                 domain=[0, 0.85],
                                 showgrid=True,
                                 range=self.axes_range[1],
                                 ),
                      xaxis2=dict(zeroline=False,
                                  domain=[0.9, 1],
                                  showgrid=True,
                                  ),
                      yaxis2=dict(zeroline=False,
                                  domain=[0.9, 1],
                                  showgrid=True
                                  ),
                      legend=dict(font=dict(size=20),
                                  bgcolor="White",
                                  itemsizing='constant'
                                  ),
                      font=dict(size=20),
                      )

    ply.plot(fig)

def plot2d_path(self):
    fig = go.Figure()

    fig.add_trace(go.Scatter(x=self.path_had[0], y=self.path_had[1],
                             mode='lines+markers',
                             marker=dict(size=1, color='red'),
                             line=dict(width=1),
                             name='2D random walk with Walsh functions',
                             )
                  )

    fig.add_trace(go.Scatter(x=self.path[0], y=self.path[1],
                             mode='lines+markers',
                             marker=dict(size=1, color='blue'),
                             line=dict(width=1),
                             name='2D regular random walk',
                             )
                  )
```

```
    fig.add_trace(go.Scatter(x=[self.path[0][0]], y=[self.path[1][0]],
                    mode='markers',
                    marker=dict(size=7, color='green'),
                    name='Start point'
                    )
            )

    fig.add_trace(go.Scatter(x=[self.path[0][-1]], y=[self.path[1][-1]],
                    mode='markers',
                    marker=dict(size=7, color='black'),
                    name='End point'
                    )
            )

    fig.update_layout(template="plotly_white",
                autosize=True,
                #width=1200,
                #height=1000,
                showlegend=True,
                xaxis=dict(showgrid=True),
                yaxis=dict(showgrid=True, scaleanchor="x", scaleratio=1),
                font=dict(size=20),
                legend=dict(font=dict(size=20),
                    bgcolor="White",
                    itemsizing='constant'
                    ),
                )
    ply.plot(fig)

def plot3d_dist(self):
    x, y, z = self.random
    x_, y_, z_ = self.random_had
    axes_range = []
    for i in self.path:
        if min(i) < 0:
            axes_range.append(2 * min(i))
        else:
            axes_range.append(-2 * min(i))
    fig = go.Figure()

    fig.add_trace(go.Scatter3d(x=x, y=y, z=z,
                    mode='markers',
                    marker=dict(size=1, color='olive'),
                    name='X-Y-Z distribution',
                    )
            )

    fig.add_trace(go.Scatter3d(x=x, y=y, z=np.ones(len(z)) * z.min() * 2,
                    mode='markers',
```

```
                    marker=dict(size=1, color='red'),
                    name='X-Y distribution',
                    hovertemplate='x:%{x:.2f}' +
                              '<br>y:%{y:.2f}<br>'
                    )
          )

fig.add_trace(go.Scatter3d(x=np.ones(len(x)) * x.min() * 2, y=y, z=z,
                    mode='markers',
                    marker=dict(size=1, color='green'),
                    name='Y-Z distribution',
                    hovertemplate='y:%{y:.2f}' +
                              '<br>z:%{z:.2f}<br>'
                    )
          )

fig.add_trace(go.Scatter3d(x=x, y=np.ones(len(y)) * min(y) * 2, z=z,
                    mode='markers',
                    marker=dict(size=1, color='blue'),
                    name='X-Z distribution',
                    hovertemplate='x:%{x:.2f}' +
                              '<br>z:%{z:.2f}<br>'
                    )
          )

fig.add_trace(go.Scatter3d(x=x_, y=y_, z=z_,
                    mode='markers',
                    marker=dict(size=1),
                    name='X-Y-Z distribution with Walsh functions',
                    visible='legendonly'
                    )
          )

fig.add_trace(go.Scatter3d(x=x_, y=y_, z=np.ones(len(z_)) * z.min() * 2,
                    mode='markers',
                    marker=dict(size=1),
                    name='X-Y distribution with Walsh functions',
                    hovertemplate='x:%{x:.2f}' +
                              '<br>y:%{y:.2f}<br>',
                    visible='legendonly'
                    )
          )

fig.add_trace(go.Scatter3d(x=np.ones(len(x_)) * x.min() * 2, y=y_, z=z_,
                    mode='markers',
                    marker=dict(size=1),
                    name='Y-Z distribution with Walsh functions',
                    hovertemplate='y:%{y:.2f}' +
                              '<br>z:%{z:.2f}<br>',
```

```
                        visible='legendonly'
                        )
            )

    fig.add_trace(go.Scatter3d(x=x_, y=np.ones(len(y)) * y.min() * 2, z=z_,
                        mode='markers',
                        marker=dict(size=1),
                        name='X-Z distribution with Walsh functions',
                        hovertemplate='x:%{x:.2f}' +
                                    '<br>z:%{z:.2f}<br>',
                        visible='legendonly',
                        )
            )

    fig.update_layout(template="plotly_white",
                    legend=dict(font=dict(size=20),
                            itemsizing='constant'
                            ),
                    )
    ply.plot(fig)

def plot3d_path(self):
    x, y, z = self.path
    x_, y_, z_ = self.path_had
    axes_range = []
    for i in self.path:
        if min(i) < 0:
            axes_range.append(2 * min(i))
        else:
            axes_range.append(-2 * min(i))
    fig = go.Figure()

    fig.add_trace(go.Scatter3d(x=x, y=y, z=z,
                        mode='lines+markers',
                        marker=dict(size=1, color='olive'),
                        name='3D random walk',
                        )
                )

    fig.add_trace(go.Scatter3d(x=x, y=y, z=np.ones(len(z)) * axes_range[2],
                        mode='lines+markers',
                        marker=dict(size=1, color='red'),
                        name='X-Y distribution',
                        hovertemplate='x:%{x:.2f}' +
                                    '<br>y:%{y:.2f}<br>'
                        )
            )

    fig.add_trace(go.Scatter3d(x=np.ones(len(x)) * axes_range[0], y=y, z=z,
```

```
                    mode='lines+markers',
                    marker=dict(size=1, color='green'),
                    name='Y-Z distribution',
                    hovertemplate='y:%{y:.2f}' +
                              '<br>z:%{z:.2f}<br>'
                    )
          )

fig.add_trace(go.Scatter3d(x=x, y=np.ones(len(y)) * axes_range[1], z=z,
                    mode='lines+markers',
                    marker=dict(size=1, color='blue'),
                    name='X-Z distribution',
                    hovertemplate='x:%{x:.2f}' +
                              '<br>z:%{z:.2f}<br>'
                    )
          )

fig.add_trace(go.Scatter3d(x=x_, y=y_, z=z_,
                    mode='lines+markers',
                    marker=dict(size=1),
                    name='3D random walk with Walsh functions',
                    )
          )

fig.add_trace(go.Scatter3d(x=x_, y=y_, z=np.ones(len(z_)) * axes_range[2],
                    mode='lines+markers',
                    marker=dict(size=1),
                    name='X-Y distribution with Walsh functions',
                    hovertemplate='x:%{x:.2f}' +
                              '<br>y:%{y:.2f}<br>'
                    )
          )

fig.add_trace(go.Scatter3d(x=np.ones(len(x_)) * axes_range[0], y=y_, z=z_,
                    mode='lines+markers',
                    marker=dict(size=1),
                    name='Y-Z distribution with Walsh functions',
                    hovertemplate='y:%{y:.2f}' +
                              '<br>z:%{z:.2f}<br>'
                    )
          )

fig.add_trace(go.Scatter3d(x=x_, y=np.ones(len(y_)) * axes_range[1], z=z_,
                    mode='lines+markers',
                    marker=dict(size=1, color='blue'),
                    name='X-Z distribution with Walsh functions',
                    hovertemplate='x:%{x:.2f}' +
                              '<br>z:%{z:.2f}<br>'
                    )
```

```
                )

        fig.update_layout(template="plotly_white",
                          legend=dict(font=dict(size=20),
                                      itemsizing='constant'
                                      )
                          )

        ply.plot(fig)



# Приклад використання
pp = Random_walk(0, 1, steps=1000, dim=3)
pp.plot()
```

ABSTRACT

Actuality of theme. In recent years, the most important trend in energy complexes around the world is the transition from outdated models of organization and management of energy complex, dominated by centralized producers, to the latest models of distributed generation, which create a more competitive environment in the energy market. At the same time, there is a strong emphasis on the use of renewable and alternative energy sources, as well as on improving the energy efficiency of existing energy infrastructure.

However, renewable energy sources are characterized by significant variability. Therefore, when developing and implementing new capacities, it is necessary to analyze the efficiency of work to improve energy efficiency, taking into account various factors, including the random component of the primary energy source.

The works of Simon Raby, Panos Argyrakis, Mark W Keller are devoted to the creation of models of random wandering.

Therefore, the topic of the dissertation research "Random walk generator with visualization is relevant.

Association of work with scientific programs, plans, topics. The dissertation was prepared in accordance with the research plan of the Department of Industrial Electronics of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute.

The aim of the study is to develop an algorithm and computer implementation of a random walk generator based on linear functions based on Walsh functions.

To achieve this goal the following **tasks** were solved:

- Review of existing models of random walk;
- Development of a mathematical model of random walk using linear functions based on Walsh functions;

- Software implementation of the developed mathematical model;
- Comparison of efficiency of models;
- Development of a startup project.

The object of research is the introduction of a mathematical model of random walk to model the random component of renewable energy sources.

The subject of the study is the choice of a simple computer-implemented model of random walk.

Research methods. The Gram-Schmidt procedure is used to solve the tasks set in the work for the development of a mathematical model of a random generator using linear functions based on Walsh functions. The computer implementation of the model was performed on a personal computer using the Python programming language.

Scientific novelty of the obtained results.

For the first time, the use of random walk to model the random component of renewable energy sources has been proposed.

For the first time, the use of orthogonal series in the random walk model has been proposed. Based on the analysis of the properties of these series, linear functions based on Walsh functions were selected, which have a relatively easy computer implementation.

The practical significance of the obtained results is to develop a model of a mathematical model using linear functions based on Walsh functions.

Personal contribution of the applicant. The dissertation is a generalization of the results of research conducted by the author himself. The work [1] was written by the author of the dissertation personally. In [2], published with co-authors, the dissertation is to build a system of linear functions based on Walsh functions using the Gram-Schmidt procedure.

Approbation of dissertation results. Theoretical provisions and results of the master's research were discussed:

1. XI International Scientific and Technical Conference of Young Scientists "Electronics-2018"

2. IV Scientific Conference of Young Scientists and Students "Generation - Transmission - Use GPW-2017"

3. V Scientific Conference of Young Scientists and Students "Generation - Transmission - Use GPW-2018"

4. VI Scientific Conference of Young Scientists and Students "Generation - Transmission - Use GPW-2019"

5. International scientific and technical conference "Problems of modern electronics - 2020"

Publications. The main provisions and results of the master's research are covered in the following publications:

1. Prediction of the wind speed change function by linear regression method, Computational problems of electrical engineering. - 2019. Vol. 9, no. 2 C. 28-33 V. Martynyuk, M, Yaremenko.

2. Martyniuk VI Linear functions based on Walsh functions / V.I. Martyniuk, KS Klen // Microsystems, Electronics and Acoustics. - 2019. - Volume 24, №1. - P. 24-39

3. Martyniuk VI, Klen KS, Zhuikov VY Modeling of cloud cover on the basis of satellite images, Proceedings of the Institute of Electrodynamics of the National Academy of Sciences of Ukraine

4. Valery Zhuikov, Vadym Martynyuk, Kateryna Osypenko The linear approximation of wind speed change function IV scientific conference of students “Generation - Transmission - Use GPW 2017”, Proceedings. - 2017 P. 31-37 (7 pages)

5. Martyniuk VI "Improving the system of Walsh functions." Proceedings of the XI International Scientific and Technical Conference of Young Scientists "Electronics-2018", p. 284-288.

The structure and scope of the dissertation. The dissertation consists of an introduction, four chapters, conclusions, a list of sources used with 51 titles and 1 appendix. The total volume of the dissertation is 106 pages, including 87 pages of the main text, 23 figures and 20 tables.