

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

Навчально-науковий інститут телекомунікаційних систем

Кафедра телекомунікацій

«На правах рукопису»

УДК _____

«До захисту допущено»

Завідувач кафедри

_____ Сергій КРАВЧУК

« ____ » _____ 2026 р.

Магістерська дисертація

на здобуття ступеня магістра

**за освітньо-науковою програмою «Інженерія інноваційних інформаційно-
телекомунікаційних технологій та систем»**

зі спеціальності 172 «Електронні комунікації та радіотехніка»

**на тему: «Метод підвищення точності позиціонування об'єктів усередині
приміщень із використанням технології Li-Fi в умовах завод»**

Виконав:

Студент VI курсу, групи ЦУ-41мн

Мікляєв Глеб Олегович _____

Керівник:

Професор кафедри ТК НН ІТС, д.т.н., професор,
Ільченко М.Ю. _____

Консультант:

Професор кафедри ТК НН ІТС, д.т.н., професор,
Романов О.І. _____

Рецензент:

Доцент кафедри ЕКІР НН ІТС, к.т.н., доцент,
Бердников О.М. _____

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних посилань.

Студент _____

Київ – 2026 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут телекомунікаційних систем
Кафедра телекомунікацій

Рівень вищої освіти – другий (магістерський)

Спеціальність – 172 «Електронні комунікації та радіотехніка»

Освітньо-наукова програма «Інженерія інноваційних інформаційно-телекомунікаційних технологій та систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Сергій КРАВЧУК

«__» _____ 2026 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Мікляєву Глебу Олеговичу

1. Тема дисертації «Метод підвищення точності позиціонування об'єктів усередині приміщень із використанням технології Li-Fi в умовах завод», науковий керівник дисертації Ільченко Михайло Юхимович, професор кафедри телекомунікацій НН ІТС, д.т.н., професор, затверджені наказом по університету від «07» квітня 2026 р. № 1415-с.

2. Термін подання студентом дисертації 06.05.2026 р.

3. Об'єкт дослідження Процес визначення координат об'єкта в приміщенні за допомогою системи Li-Fi в умовах наявності завод

4. Предмет дослідження Модель та метод визначення місцеположення об'єкта в кімнаті засобами Li-Fi при наявності значних завод на певній кількості передавачів, зокрема методами RSS та Fingerprint.

5. Перелік завдань, які потрібно розробити:

1. Розробити базу даних опорних точок Fingerprint для RSS методу технології Li-Fi

2. Модифікувати метод позиціонування із використанням технологією Li-Fi для можливого пошуку об'єкта за двома лампами

3. Розробити модель для покращеного методу позиціонування із використанням технологією Li-Fi в умовах завод

4. Провести імітаційне моделювання визначення місцеположення об'єктів в умовах, коли сигнал лише від 2 ламп прийнято без спотворень

5. Проаналізувати отримані результати та розробити рекомендації щодо використання розробленого методу та подальших досліджень

6. Орієнтовний перелік ілюстративного матеріалу

1. Тема, мета, актуальність задачі дослідження

2. Аналіз існуючих методів внутрішнього позиціонування сумісних із технологією Li-Fi

3. Запропонована модель та метод внутрішнього Li-Fi позиціонування

4. Результати імітаційного моделювання

5. Висновки

7. Орієнтовний перелік публікацій

1. Романов, О. І. ., & Мікляєв, Г. О. . (2025). ПРОГНОЗУВАННЯ СПОТВОРЕНИХ СИГНАЛІВ ВІД LED ЛАМП В СИСТЕМАХ ПОЗИЦІОНУВАННЯ ВСЕРЕДИНІ ПРИМІЩЕНЬ. *Збірник матеріалів Міжнародної науково-технічної конференції «ПЕРСПЕКТИВИ ТЕЛЕКОМУНІКАЦІЙ»*, 120–122. вилучено із <https://conferenc-journal.its.kpi.ua/article/view/331874>

2. Романов, О. І. ., & Мікляєв, Г. О. . (2026). СТВОРЕННЯ БАЗИ ДАНИХ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ ВНУТРІШНЬОГО ПОЗИЦІОНУВАННЯ МЕТОДОМ "FINGERPRINT". *Збірник матеріалів Міжнародної науково-технічної конференції «ПЕРСПЕКТИВИ ТЕЛЕКОМУНІКАЦІЙ»*, 139–141.

3. Мікляєв Г.О. Рішення задачі позиціонування всередині приміщень на базі комплексного використання методів Fingerprint та RSS, Всеукраїнський конкурс студентських наукових робіт з галузей знань і спеціальностей у 2025/2026 н.р.

4. Romanov, O., & Miklaiv, G. (2021). THROUGHPUT INCREASE IN THE OFFICE LI-FI SYSTEM. *Information and Telecommunication Sciences*, 12(2), 16–21. <https://doi.org/10.20535/2411-2976.22021.16-21>

8. Консультанти розділів дисертації*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Романов О.І. професор		

□ Якщо визначені консультанти. Консультантом не може бути зазначено наукового керівника магістерської дисертації.

9. Дата видачі завдання “2 ” вересня 2024 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Пошук та аналіз літератури на тематику позиціонування Li-Fi	02.09.2024 - 30.09.2024	виконано
2	Модифікація алгоритму позиціонування із технологією Li-Fi для можливого пошуку об'єкта за двома лампами	01.10.2024 - 31.10.2024	виконано
3	Розробка програми створення загальної бази даних	01.11.2024 - 30.11.2024	виконано
4	Розробка програми визначення ламп, сигнал від яких спотворено	01.12.2024 - 24.12.2024	виконано
5	Розробка програми визначення допоміжних точок	01.02.2025 - 28.02.2025	виконано
6	Розробка програми пошуку об'єкта із використанням допоміжних точок	01.03.2025 - 31.03.2025	виконано
7	Дослідження кількості потенційних точок місцеположення об'єкта при пошуку за двома світлодіодними лампами в системі Li-Fi	01.04.2025 - 30.04.2025	виконано
8	Дослідження відстані між потенційними точками місцеположення об'єкта при пошуку за двома світлодіодними лампами в системі Li-Fi	01.05.2025 - 31.05.2025	виконано
9	Дослідження правильності визначення ламп, сигнал від яких не є спотвореним	01.09.2026 - 30.09.2026	виконано
10	Дослідження доцільності використання пари ламп, серед яких сигнал від однієї лампи прийнято із спотвореннями	01.10.2026 - 31.10.2026	виконано
11	Дослідження частки помилок при визначенні місцеположення в залежності від взаєморозташування ламп, сигнал від яких спотворено	01.11.2026 - 30.11.2026	виконано
12	Дослідження частки помилок при визначенні місцеположення в залежності від значущості спотворення	01.12.2026 - 24.12.2026	виконано
13	Дослідження залежності правильності визначеного місцеположення в залежності від розташування ламп на стелі	01.02.2026 - 28.02.2026	виконано
14	Дослідження доцільності використання алгоритму для пошуку допоміжних точок, розробленого в бакалаврській роботі	01.03.2026 - 30.03.2026	виконано
15	Аналіз результати роботи	01.04.2026 - 30.04.2026	виконано
16	Оформлення результатів роботи	01.05.2026 - 06.05.2026	виконано

Студент

Глеб МІКЛЯЄВ

Науковий керівник дисертації

Михайло ІЛЬЧЕНКО

РЕФЕРАТ

Магістерська дисертація містить 117 сторінок, 10 таблиць, 47 рисунків, 10 формул, 2 додатки. Перелік посилань нараховує 73 найменувань.

Актуальність теми: Із збільшенням кількості пристроїв та об'єму передаванні даних набуває актуальність використання технології Li-Fi, яка має у 10000 разів вищу потенційну пропускну здатність за Wi-Fi. Li-Fi також має вищу точність, При використанні Li-Fi для позиціонування спотворення на одному з передавачів може сильно вплинути на результати позиціонування. а даний час залишається невирішеною проблема завад в Li-Fi системах. Завада до сигналу може бути як додатною, так і від'ємною. При взаємодії із додатною завадою, прийнятий сигнал є збільшеним. Така завада може виникати внаслідок потрапляння сонячних променів, відзеркалених променів, при перегоранні резисторів на лампі або від інших джерел освітлення. При взаємодії із від'ємною завадою прийнятий сигнал згасає. Вона виникає через проходження напівпрозорі поверхні або через перегорання світлодіодів на передавачі чи помилці при синхронізації.

Зв'язок роботи із науковими програмами, планами та темами.

Магістерську дисертацію на здобуття ступеня магістра виконано в Національному технічному університеті України «КПІ імені Ігоря Сікорського» (м. Київ) відповідно до планів науково-дослідних робіт кафедри Телекомунікацій.

Метою дослідження є підвищити точність визначення місцеположення об'єкта в приміщенні за рахунок розроблення метода визначення місцеположення об'єкта в Li-Fi системі завдяки аналізу потужності світлового потоку від світлодіодних ламп.

Об'єкт дослідження: Процес визначення координат об'єкта в приміщенні за допомогою системи Li-Fi в умовах наявності завад.

Предмет дослідження: Модель та метод визначення місцеположення об'єкта в кімнаті засобами Li-Fi при наявності значних завад на певній кількості передавачів, зокрема методами RSS та Fingerprint.

Методи дослідження: основними методами дослідження є математичне та імітаційне моделювання.

Наукова новизна одержаних результатів.

У роботі вперше створено метод, який дозволяє визначити місцеположення об'єктів в умовах завад, у результаті яких лише від двох передавачів сигнал прийнято без спотворень шуканим об'єктом із використанням технології Li-Fi.

Практичне значення одержаних результатів.

Результати дослідження дозволяють підвищити точність визначення місцеположення об'єктів в умовах завад використовуючи технологію Li-Fi

Апробація результатів дисертації.

Результати дослідження представлено на міжнародних конференціях «Перспективи телекомунікацій 2025» та «Перспективи телекомунікацій 2026», на «Всеукраїнському конкурсі студентських наукових робіт з галузей знань і спеціальностей у 2025/2026 н.р.»

Ключові слова: *Li-Fi, positioning, indoor positioning, RSS, Fingerprinting, VLC, Visible Light Communication, Received Signal Strength.*

ABSTRACT

Master's thesis comprises 117 pages, 10 tables, 47 figures, 10 formulas and 2 appendices. The reference list contains 73 entries.

Relevance of the topic:

With the increasing number of devices and the growing volume of data being transmitted, the use of Li-Fi technology is becoming increasingly relevant, as it has a potential bandwidth 10,000 times greater than that of Wi-Fi. Li-Fi also offers greater accuracy; however, when using Li-Fi for positioning, a distortion in one of the transmitters can significantly affect the positioning results. Furthermore, the issue of interference in Li-Fi systems remains unresolved. Interference with the signal can be either positive or negative. When interacting with positive interference, the received signal is amplified. Such interference may arise from direct sunlight, reflected rays, burnt-out resistors in a lamp, or other light sources. When interacting with negative interference, the received signal is attenuated. This occurs due to transmission through a semi-transparent surface, the burnout of LEDs on the transmitter, or a synchronisation error.

Relationship to research programmes, plans and topics.

This Master's thesis was completed at the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" (Kyiv) in accordance with the research plans of the Department of Telecommunications.

The purpose of study is to improve the accuracy of indoor object localisation by developing a method for determining the indoor location of an object in a Li-Fi system through the analysis of the luminous flux from LED lamps.

Object of research:: The process of determining the coordinates of an object within a room using a Li-Fi system in the presence of interference.

Subject of research:: A model and method for determining the location of an object within a room using Li-Fi in the presence of significant interference on a certain number of transmitters, specifically using RSS and Fingerprint methods.

Methods of research:: The main research methods are mathematical and simulation modelling.

Scientific novelty of the obtained results:

This paper presents, for the first time, a method that enables the location of objects to be determined in the presence of interference, whereby the target object receives undistorted signals from just two transmitters using Li-Fi technology.

The practical value of the results obtained:

The results of the study make it possible to improve the accuracy of object localisation in the presence of interference using Li-Fi technology

Results approbation:

The research findings were presented at the international conferences ‘ “Prospects for development of information-telecommunication technologies and systems 2025” and ‘Prospects for development of information-telecommunication technologies and systems 2026” , as well as at the "Всеукраїнського конкурсу студентських наукових робіт з галузей знань і спеціальностей у 2025/2026 н.р."

Keywords: *Li-Fi, positioning, indoor positioning, RSS, Fingerprinting, VLC, Visible Light Communication, Received Signal Strength.*

ЗМІСТ

ВСТУП.....	12
РОЗДІЛ 1	14
ТЕОРЕТИЧНІ ОСНОВИ ВНУТРІШНЬОГО Li-Fi ПОЗИЦІОНУВАННЯ.....	14
1.1 Актуальність та шляхи вирішення задачі внутрішнього позиціонування	14
1.2 Актуальність технології Li-Fi, принцип роботи та компоненти Li-Fi системи	19
1.3 Огляд методів внутрішнього позиціонування	29
Висновки	36
РОЗДІЛ 2	38
РОЗРОБКА МОДЕЛІ ТА МЕТОДУ ВИРІШЕННЯ ЗАДАЧІ ВНУТРІШНЬОГО ПОЗИЦІОНУВАННЯ ПРИ НАЯВНОСТІ ЗАВАД	38
2.1 Аналіз методів.....	38
2.2 Опис структури мережі Li-Fi та Li-Fi моделі	39
2.3 Методика вирішення задачі визначення місцеположення об'єкта	44
2.3.1 Складання загального алгоритму	44
2.3.2 Алгоритм створення і заповнення бази даних відбитку пальців (F1).	52
2.3.3 Алгоритм визначення не спотворених ламп і потенційних координат місцеположення об'єкта (F2)	57
2.3.4 Визначення допоміжних точок (F3)	64
2.3.5 Визначення координати об'єкта серед потенційно можливих точок із використанням допоміжних точок (F4).....	66
Висновки	71
РОЗДІЛ 3	73
РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ ВНУТРІШНЬОГО ПОЗИЦІОНУВАННЯ ПРИ НАЯВНОСТІ ЗАВАД	73
3.1 Створення бази даних	73
3.2 Визначення ламп, сигнал від яких не спотворений	76
3.3 Визначення координат об'єкта, із використанням інформації про спотвореність сигналу від ламп	86
Висновки	92
ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ.....	94
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	96
ДОДАТОК А ЛІСТИНГ ПРОГРАМИ.....	106
А.1 Лістинг основного коду	106
А.2 Лістинг налаштувань конфігурації	112
ДОДАТОК Б ПОЧАТКОВІ ПАРАМЕТРИ	114

ПЕРЕЛІК СКОРОЧЕНЬ

VLC	Visible Light Communication	Комунікація за допомогою видимого світла
AOA	Angle of Arrival	Кут приходу
RSS	Received Signal Strength	Потужність прийнятого сигналу
RSSI	Received Signal Strength Indicator	Показник потужності прийнятого сигналу
TOA	Time of Arrival	Час приходу
TDOA	Time Difference of Arrival	Різниця часу приходу

ВСТУП

Зараз продовжується збільшення автоматизації та зменшення залученості людини. І зі збільшенням автоматизації постійно збільшується складність задач, які машини виконують. На даний час роботи і пристрої інтернет речей потребують визначення місцеположення об'єктів в системі, для виконання автоматизованих задач. Особливо важливим є визначення місцеположення при використанні роботів, це може бути як в лікарнях, наприклад, для автоматизованого руху ліжок, так і на складах, для автоматичного розвантаження та завантаження товарів. Для цього потрібно використання деяких технологій. Серед традиційних методів, таких як Wi-Fi, ZigBee, Bluetooth є багато недоліків, такі як мала точність, проблеми з електромагнітною сумісністю серед інших радіопристроїв, тощо. Окрім вищезазначеного, рішення для зовнішнього позиціонування, такі як системи супутникової навігації, мають занадто слабкий сигнал в приміщеннях через значне згасання при проходженні крізь приміщення. Технологія Li-Fi, в свою чергу, не має перелічених недоліків і є дуже актуальним в цій області. Також Li-Fi має ряд переваг, як от потенційна більша пропускна здатність у 1000 разів більша за Wi-Fi. Ця технологія не створює завад та не вразлива до них. Дообладнання Li-Fi систем потенційно значно дешевше за Wi-Fi системи. Це переваги роблять цю технологію дуже перспективною, тому прогнозується, що ця технологія матиме дуже широке застосування у майбутньому. Передавачами в технології Li-Fi є світлодіодні лампи, а приймачами – фотодетектори, які широко використовуються в нас час в повсякденному житті. Хоча сигнал Li-Fi стійкий до радіозавад, світлові завади для світлодіодних приймачів в системі Li-Fi залишаються проблемою для позиціонування. На даний час залишається невирішеною проблема завад в Li-Fi системах. Завада до сигналу може бути як додатною, так і від'ємною. При взаємодії із додатною завадою, прийнятий сигнал є збільшеним. Така завада може виникати внаслідок потрапляння сонячних променів, відзеркалених променів, при перегоранні резисторів на

лампи або від інших джерел освітлення. При взаємодії із від'ємною завадою прийнятий сигнал згасає. Вона виникає через проходження напівпрозорі поверхні або через перегорання світлодіодів на передавачі чи помилці при синхронізації. Тому приділено окрему увагу розробці алгоритму, який можливий при використанні не лише в ідеальних умовах, а ще й при наявності певних завад. Аналогічні наукові дослідження не розглядають визначення координат позиціонування в умовах завад. Тому ця робота присвячена дослідженню визначення місцеположення об'єктів за допомогою технології Li-Fi в умовах завад.

РОЗДІЛ 1

ТЕОРЕТИЧНІ ОСНОВИ ВНУТРІШНЬОГО LiFi ПОЗИЦІОНУВАННЯ

1.1 Актуальність та шляхи вирішення задачі внутрішнього позиціонування

Зараз відбувається четверта індустріальна революція і збільшення кількості інтернет речей (IoT). Є потреба в визначенні їх місцезнаходження для взаємодії між пристроями. Ззовні та на великих місцевостях добре працює GPS, але всередині приміщень його функціонування не є достатньо необхідним. Як варіант рішення пропонується в роботі використовувати позиціонування Li-Fi. Він може використовуватися для приміщень різних типів та для вирішення різних підзадач у позиціонуванні у логістиці та управлінні запасами на складах (оптимізація маршрутів, пошук товарів), навігації всередині будівель, лікарні (пошук медичного обладнання, керування руху лікарняних ліжок), тощо[1].

У контексті промислових Інтернет-речей, високоточне і надійне позиціонування всередині приміщень має вирішальне значення для нових додатків, які мають можливості визначення місцезнаходження. Типові сценарії застосування включають відстеження людей і товарів на інтелектуальних фабриках, позиціонування і навігація автоматизованих керованих транспортних засобів (AGV), а також керування автомобілями на підземних паркінгах. Через складне внутрішнє середовище, традиційні методи позиціонування в приміщенні, які використовують радіочастотні (РЧ) бездротові сигнали в приміщенні (в тому числі WiFi і Bluetooth), схильні до багатопроменевого поширення сигналу, що призводить до поганої оцінки положення[2].

Позиціонування – це важливий інструмент для надання різних послуг, таких як навігація, створення карт, відстеження об'єктів. Головною технологією позиціонування є GPS, яка забезпечує геолокацію. Проте, GPS менш точний у приміщеннях через перешкоди, такі як стіни. Тому розвинули систему

позиціонування IPS на основі бездротових технологій, таких як Li-Fi, Wi-Fi і Bluetooth[3].

На Рис. 1.1 зображений роботизований склад, в якому як рухомим, так і нерухомим роботам потрібне визначення місцеположення об'єктів і своє власне. Рухомим задля безпечного та правильного руху, нерухомим для безпечного та правильного знаходження товару та місця, на яке потрібно поставити товар[3].



Рис. 1.1 Високороботизований склад із потребою в позиціонуванні [4]

Дослідники все більше застосовують для високоточного позиціонування в приміщеннях разом із технологію Інтернету речей (IoT). Стандартні технології позиціонування, такі як радіочастотна ідентифікація (RFID), Wi-Fi і Bluetooth, мають обмеження у забезпеченні точного позиціонування через перешкоди та ефекти відбивання сигналів у приміщенні. [4,5]

UWB SPECTRUM

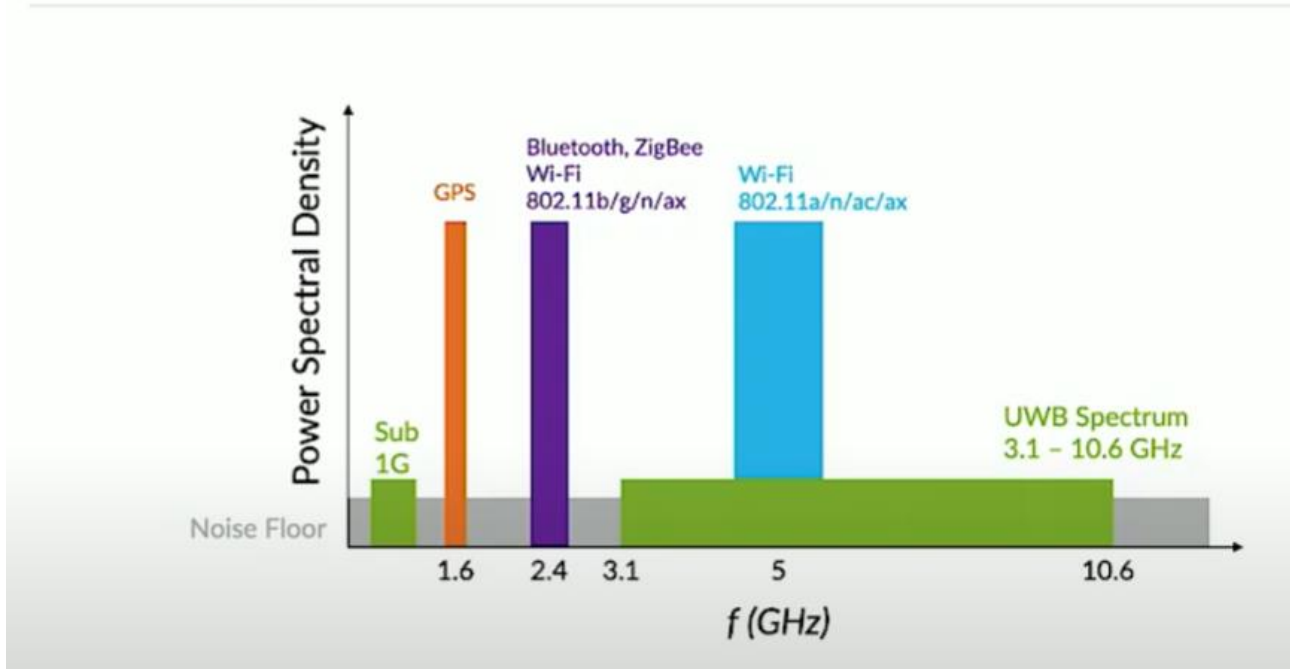


Рис. 1.2 Потужнісний спектральний розподіл частоти різних радіотехнологій в залежності від частоти [5, 6].

Технологія Ultra Wide Band має вищу точність позиціонування та вищу пропускну здатність, ніж Wi-Fi, ZigBee, Bluetooth, але ця технологія може працювати лише у ізоляції від інших радіотехнологій та дуже чутлива до будь-яких радіозавад, тому її важко запровадити. На Рис. 1.2 зображено спектр використання UWB із порівнянням інших радіотехнологій. Окрім цього, вона використовує значну кількість додаткового обладнання, такого як якорі: Стационарні пристрої, встановлені за відомими координатами всередині будівлі (аналогічно супутникам GPS), мітки: мобільні пристрої, прикріплені до особи або об'єкта, що відстежується (наприклад, смартфон або носимий датчик), яке, в свою чергу дороге. Ця технологія споживає багато енергії. [6,7]



Рис. 1.3. Приклад розумного госпіталю із рухомими ліжками [8,9]

На Рис. 1.3 зображено приклад госпіталю із рухомими ліжками, у якому використання радіосигналів обмежене, через що використання технологій Wi-Fi, Bluetooth, UWB, ZigBee не підходить. [10]

Цей метод містить також кілька інших недоліків: Обладнання має визначати або часову затримку у долі наносекунд, або кут приймання, враховуючи можливі коливання. Через що ця технологія є найдорожчою. Окрім того, ця технологія потребує прямої видимості: Хоча імпульси UWB можуть проникати крізь деякі матеріали, стіни з важких металів або товстого бетону можуть відбивати або затримувати сигнал, що призводить до помилок «багатопрореневого поширення». Також має бути принаймні 4 опорні точки на кімнату з особливими вимогами геометричного розміщення (не розташовані на одній прямій або в одній площині), щоб уникнути значного зниження точності (DOP, Dilution of Precision). Дальність технології складає не більше десятків метрів та нестійка до завад [11]

Технологія ультразвукового позиціонування також є порівняно точна,, але потребує додаткового обладнання, що підвищує вартість реалізації та впровадження технології. На відмінно від технологій позиціонування, заснованих на вимірюванні часу проходження світла, ультразвук повільніше на 6 порядків, тому замість часток наносекунд точність має складати у часті мілісекунд. З іншого боку, це також є проблемою: Швидкість звуку залежить від багатьох різних факторів, таких як коливання температури, тиску в приміщенні. Врахування цих факторів значно поскладнює та здорожчує систему. Хоча ця технологія стійка для електромагнітних завад, нестійка до ультразвукових. Багато машин створює шум, який глушить корисний сигнал, через що використання даної технології стає неможливою. Також ультразвук не проходить крізь непрозорі предмети. [12]

Є дві найпоширеніші архітектури: Активна (bat) та пасивна (cricket). В активній мобільний датчик випромінює ультразвуковий імпульс, який вловлюють стаціонарні датчики, встановлені на стелі. Це добре підходить для відстеження великої кількості об'єктів, але може швидко розряджати акумулятор датчика. В пасивній стаціонарні маяки одночасно випромінюють ультразвукові та радіочастотні сигнали. Мобільний пристрій обчислює своє місцезнаходження на основі різниці в часі між «світлом» (радіочастотним сигналом) і «звуком» (ультразвуковим сигналом). Це забезпечує кращий захист приватних даних, оскільки пристрій не передає інформацію про своє місцезнаходження. [13]

Позиціонування із використанням VLC має багато переваг, такі як низька вартість на дообладнання. Готова Li-Fi система вже є готовою інфраструктурою для позиціонування. Ствітловий промінь є набагато точнішою за радіотехнології, також є сумісним із іншими радіотехнологіями, не створює завади у радіодіапазоні та не вразливий до радіозавад. Серед недоліків варто зазначити потреба у прямому промені та створення та вразливість до завад у світловому діапазоні. [14]

1.2 Актуальність технології Li-Fi, принцип роботи та компоненти Li-Fi системи

На сьогодні одна з найбільш поширених технологій безпроводового доступу є технологія Wi-Fi, яку застосовують для підключення комп'ютерів, мобільних телефонів, планшетів та інших пристроїв до Інтернету. Також Wi-Fi широко використовують у громадських місцях, таких як метро, гіпермаркети, стадіони, торговельно-розважальні центри аеропорти, майдани, в офісах компаній і фірм. Але користувачів стає все більше, зростають вимоги до пропускної спроможності систем і разом із тим збільшують радіочастотні завади. Як показує прогноз, наданий компанією CISCO, і попереджує Федеральна комісія радіозв'язку США, Wi-Fi підходить до межі своїх можливостей з пропускної здатності. Крім того, є обмеження на використання радіохвиль в літаках, лікарнях та інших місцях[7,15].

Тому, в якості одного із можливих напрямків подальшого розвитку систем безпроводового доступу, групою вчених з Единбурзького університету під керівництвом професора Гаральда Хааса запропонована технологія Li-Fi. Li-Fi – це безпроводова оптична система зв'язку, в якій для передавання даних замість радіохвиль використовують світло. Різна амплітуда і швидкість мерехтіння світла дає змогу передавати дані. Інтенсивність мерехтіння дуже висока, тому людське око його не помічає та не викликає стимулів під час епілепсії. Технологія Li-Fi нова, тому всі системи на базі технології працюють поки ще в лабораторних чи експериментальних умовах. Однак останнім часом, на базі технології Li-Fi пропонують абсолютно новий підхід до побудови систем безпроводового доступу щодо швидкості передавання даних, гнучкості та комфортності експлуатації і починають створювати зразки, які можливо використовувати на практиці[8, 16].

Розглянемо переваги та недоліки технології Li-Fi, загалом та у порівнянні до технології Wi-Fi:

Переваги технології Li-Fi

1. Діапазон технології Li-Fi складає до 100'000 разів більше за канал радіозв'язку, який, наприклад, викривується у Wi-Fi та у 1000 разів більше за увесь радіодіапазон, у якому розміщується безліч сучасних технологій, як зображено на Рис. 1.4. Оскільки Li-Fi не перевантажений сигналами від мобільних телефонів, мікрохвильових печей та телевізорів, ця технологія дозволяє досягти значно вищої щільності передачі даних. Через що потенційна пропускна здатність може досягати петабіт на секунду. Хоча це потребує додаткових досліджень, покращення програмних засобів та апаратного забезпечення, розробки та впровадження нових систем. [17]

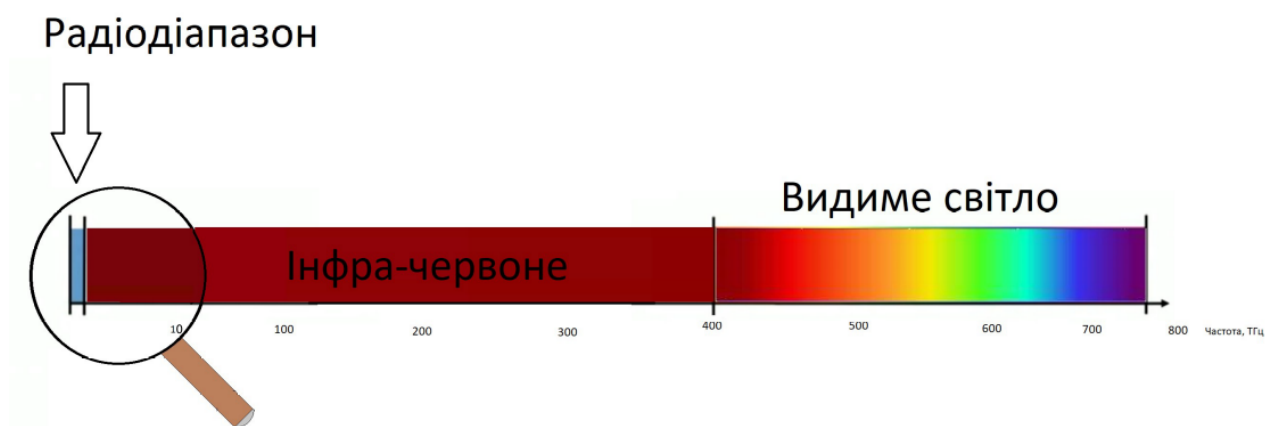


Рис. 1.4 Лінійне порівняння радіодіапазону та видимого світла

2. Впровадження технології Li-Fi можливе використання вже побудованої мережі освітлення приміщення. Li-Fi використовує існуючу інфраструктуру світлодіодного освітлення. Оскільки майже усі будівлі перейшли на світлодіодне освітлення (LED) з метою економії енергії, ці самі лампи можна адаптувати для використання в якості точок доступу до даних. Світлодіодна лампа забезпечує як освітлення, так і передачу даних, що зменшує потребу в додаткових маршрутизаторах та обладнанні. Серед додаткових елементів це LED Driver та Li-Fi контролер на мережу. Це скорочує витрати на побудову мережі передавання даних та позиціонування. У «розумному» будинку чи офісі

кожна лампочка стає потенційним високошвидкісним передавачем, що гарантує відсутність «мертвих зон». [18]

3. Обмежена доступність використання радіохвиль, наприклад, неможливе використання в літаках та лікарнях, на ядерних реакторах та в інших місцях, де вже наявне інтенсивне радіовипромінення. Технологія Li-Fi ніяк не взаємодіє із радіохвилями, тобто вона не створює їх, а їх наявність чи відсутність не впливає на роботу системи. Інтерференція повністю відсутня. Далі розглянуто кілька детальніших прикладів [19]

У лікарнях: це не спричиняє завад у роботі чутливого медичного обладнання або апаратів МРТ.

У авіація: пасажери можуть користуватися Li-Fi під час польоту, не впливаючи на системи зв'язку літака.

На промислових підприємствах: безпечно використовувати в умовах, де радіочастотні сигнали можуть спричинити завади в роботі спеціалізованого обладнання. [19]

4. Вища ефективність у Li-Fi за Wi-Fi [10]:

У Wi-Fi базові станції споживають більше енергії ніж в технології Li-Fi; для збільшення передавання даних потрібно збільшувати спектр частот, а радіодіапазон перевантажений; [11, 20]

Обмежена потужність, великі витрати на живлення. Його використовують тільки для передавання даних та позиціонування, у порівнянні до технології Li-Fi, яка будується на інфраструктурі освітлення тому великі затрати [9, 20].

5. Вищий захист інформації:

Радіохвилі проходять крізь стіни, а це означає, що сигнал Wi-Fi може бути перехоплений кимось, хто перебуває за межами кімнати чи будівлі. Натомість світло не проходить крізь непрозорі конструкції. [21]

Оскільки світло не може проникати крізь стіни, сигнал Li-Fi фізично обмежується приміщенням, у якому світиться світло, створюючи, фактично,

фізичний брандмауер. Така ізоляція запобігає атакам типу «людина посередині» ззовні, що робить цю технологію кращим вибором для середовищ з високим рівнем безпеки, таких як військові об'єкти, банки, лікарні державні установи та інші [22]

6. Екологічність

Система на базі технології Li-Fi є набагато екологічнішою за wi-fi систему тому що використовує менше компонентів. Як вже зазначалося у п.1, для Li-Fi системи потрібно лише контролер та LED драйвери, які під'єднуються до світлодіодних ламп, які є передавачами у Li-Fi системах. Під час виробництва системи відбувається менше викидів вуглекислого газу та інших газів які належать до інших паркових газів, що викликають глобальне потепління та газів, які забруднюють атмосферу. Також ці компоненти нескладні у переробці, утилізації. [23]

7. Безпека для здоров'я людини

комунікація за допомогою видимого світла є безпечнішою для людини за радіотехнології. Воно не проходить крізь шкіру, не нагріває організм та не викликає інших негативних ефектів для організму людини. Мерехтіння не є небезпекою, оскільки воно занадто швидке, відбувається із періодом меншим за наносекунду, для того щоб людський організм міг сприйняти це за мерехтіння і натомість він приймає це як суцільне світло. Мінімальна частота у гігареци незрівнянно більше за сучасні монітори. Використання додаткових алгоритмів забезпечує стабільне освітлення без великої черги нулів та одиниць які могли б вплинути негативно на людину. Отже, Li-Fi не є шкідливим для людей із епілепсією. Він не створює додаткового впливу на організм за той, що вже наявна від освітлення у кімнаті. [24]

8. Використання неліцензованого спектру^[2, 3, 13]. Тобто не потрібно сплачувати додатково за ліцензію. Оскільки промені світла не покидають приміщення, на відміну від радіотехнологій, не потрібно обмежувати їх використання. Хоча питання майбутнього ліцензування VLC технологій

залишається відкритим, зрозуміло точно, що воно не буде настільки дорогим та закритим як із радіоспектром. [25]

9. Як зазначено у п.1, Li-Fi має значно більшу частоту, а отже і коротшу довжину хвилі, через що максимальна потенційна точність під час позиціонування набагато більша. Хоча це потребує додаткових досліджень, покращення програмних засобів та апаратного забезпечення, розробки та впровадження нових систем. [26]

Недоліки та виклики технології Li-Fi

1. Фізичні перешкоди та дальність дії

Найбільш нагальною проблемою є те, що світло не може проникати крізь тверді предмети, такі як стіни. Хоча це робить мережу надзвичайно захищеною від прослуховування, це також обмежує зону дії до однієї кімнати. Для стабільного з'єднання приймач зазвичай потребує прямого або майже прямого шляху до джерела світла. Li-Fi зазвичай ефективний лише в радіусі близько 10 метрів, що вимагає щільної мережі світлодіодних точок доступу. [27]

2. Вплив навколишнього освітлення

Технологія Li-Fi працює за рахунок модуляції інтенсивності світла зі швидкістю, невидимою для людського ока. Однак зовнішні джерела світла можуть спричиняти «шум» у сигналі. Основний джерелом таких шумів є Сонячне світло. Прямі сонячні промені містять широкий спектр світла, який може перенасичувати приймачі, ускладнюючи датчику розрізнення мерехтіння, що несе дані, від фонового освітлення. Також високоінтенсивне внутрішнє освітлення від інших джерел може заважати чіткості сигналу, що призводить до втрати пакетів або зниження швидкості. [28]

3. Проблема висхідного каналу

Більшість сучасних Li-Fi пристроїв та експериментів у лабораторіях зосереджуються на низхідному каналі (від світлодіодних ламп до пристрою користувача). Створення ефективного висхідного каналу (від пристрою до

мережі) є більш складним завданням. Мобільні пристрої повинні бути оснащені достатньо потужними випромінювачами інфрачервоного або видимого світла, щоб надсилати дані назад до датчиків на стелі, але постійне випромінювання світла зі смартфона або ноутбука для підтримання інтернет-з'єднання може значно скоротити час роботи акумулятора. Також поряд із світлодіодними лампами має бути розміщені фотоприймачі висхідного каналу, що збільшує вартість дообладнання інфраструктури та негативний екологічний вплив. Забезпечення прямого вихідного променя складно забезпечити із мобільних рухомих пристроїв. Також це може бути дратівливим для користувачів. Використання однонаправленої мережі Li-Fi із іншим зворотнім каналом частково вирішує цю проблему, але створює недоліки від технології, яка забезпечує зворотнім каналом. [29]

4. Постійно увімкнені ліхтарі:

Щоб Інтернет працював, ліхтарі мають бути увімкнені. Хоча їхню яскравість можна зменшити до рівня, при якому вони здаються людині «вимкненими», але при цьому продовжують передавати дані, вони мають залишатися увімкненими цілодобово. Для вирішення цієї проблеми можна використовувати інфрачервоне світло або близький ультрафіолет, який непомітний людському оку. [30]

5. Проблеми із змінною передавача

Перехід з однієї кімнати в іншу вимагає безперебійної передавання між різними лампочками. Сучасні мережеві протоколи не можуть забезпечити перемикання між цими невеликими «клітинками» без переривання потоку даних. Вирішення цієї проблеми лежить у майбутніх дослідженнях. [31]

6. Розміри обладнання

Наразі приймачі та передавачі Li-Fi зазвичай мають вигляд зовнішніх адаптерів або громіздких компонентів. Інтеграція цих датчиків у тонкі корпуси сучасних смартфонів та смарт-годинників без погіршення якості камери чи зменшення розміру екрана залишається викликом для виробників. [32]

На Рис. 1.5 можна побачити приклад кімнати, в якій використовується система Li-Fi з Li-Fi контролером, двома LED-лампами і чотирма кінцевими пристроями. Помаранчевим пунктиром зображено світловий канал. [33]

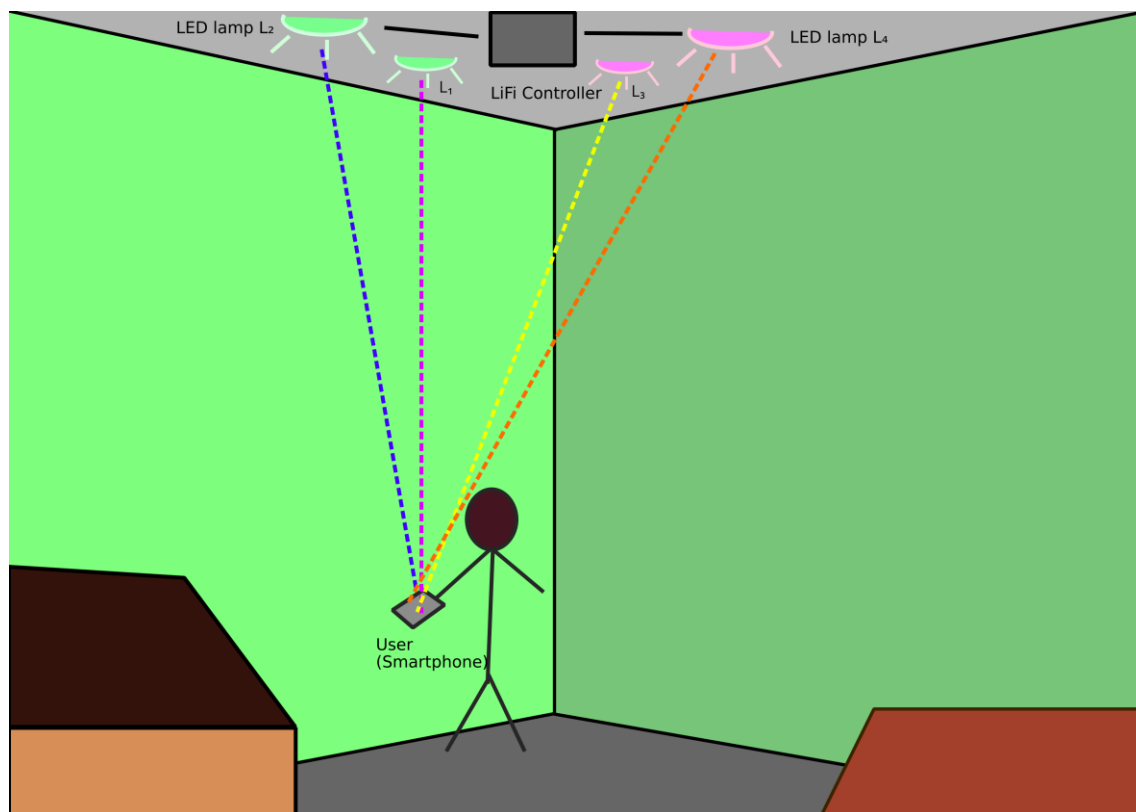


Рис. 1.5 Кімната з Li-Fi

Основна ідея VLC, який використовує Li-Fi, полягає у додаванні інформаційних даних до змін інтенсивності світла та виявлення змін на стороні приймача. Це називають методом прямого виявлення, модульованого інтенсивністю intensity modulated direct detected (IM/DD). Використовувана частота модуляції зазвичай набагато вища, ніж може виявити людське око, і тому це не спричинює помітного мерехтіння в освітленні[12,16-17].

Основними елементами мережі LI-FI є: світлодіодний драйвер, LED лампа, фотоприймач.

Світлодіодний драйвер забезпечує управління LED лампами (Рис 1.6).



Рис. 1.6 Структура світлодіодного драйвера - Li-Fi Driver

Світлодіодним драйвером є електричний пристрій, який регулює живлення світлодіоду або рядка світлодіодів. Є кілька різних видів світлодіодних драйверів. Найбільш поширені для живлення від 2,3В до 5,5 В, від 2,7 В до 5,5 В і від 3 В до 5,5 В. [34]

Другий елемент, це LED лампа. На Рис.1.7 показано принцип дії світлодіодної лампи LED.

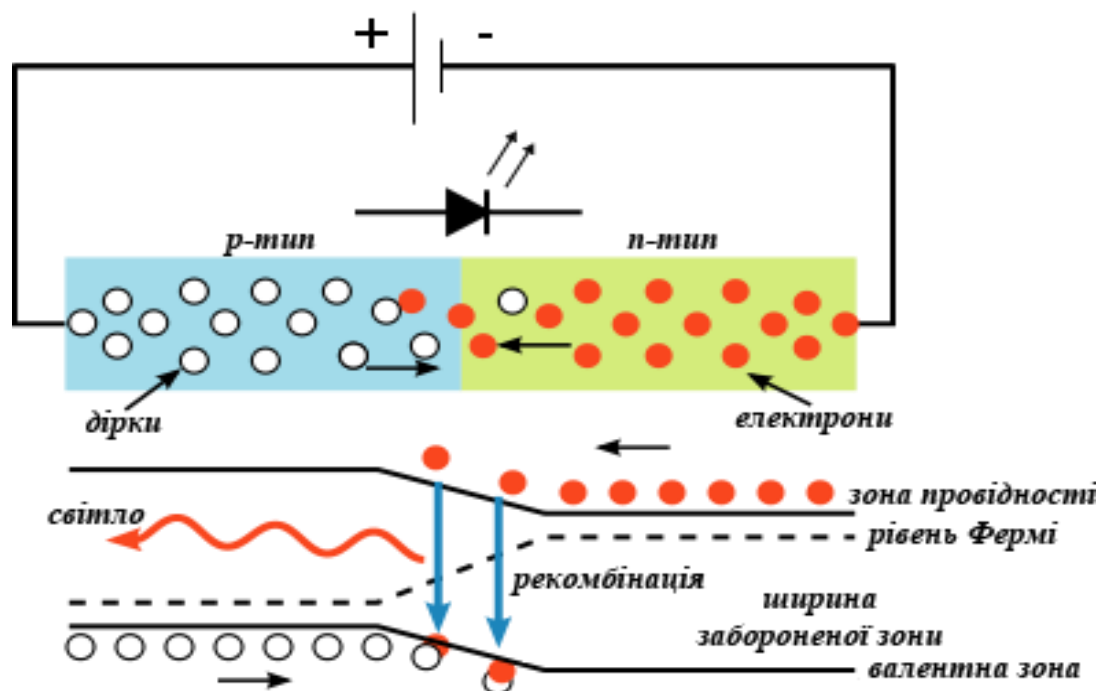


Рис. 1.7 Принцип дії світлодіодної лампи

Робота світлодіодної лампи основана на інжекції електронів під час проходження струму через діод. Процес рекомбінації інжекттованих електронів існує, і в базовій області, і в р-п переході. Це супроводжується переходом електронів з вищого енергетичного рівня на нижчий. Приклад Li-Fi передавача зображено на Рис. 1.8 [35]

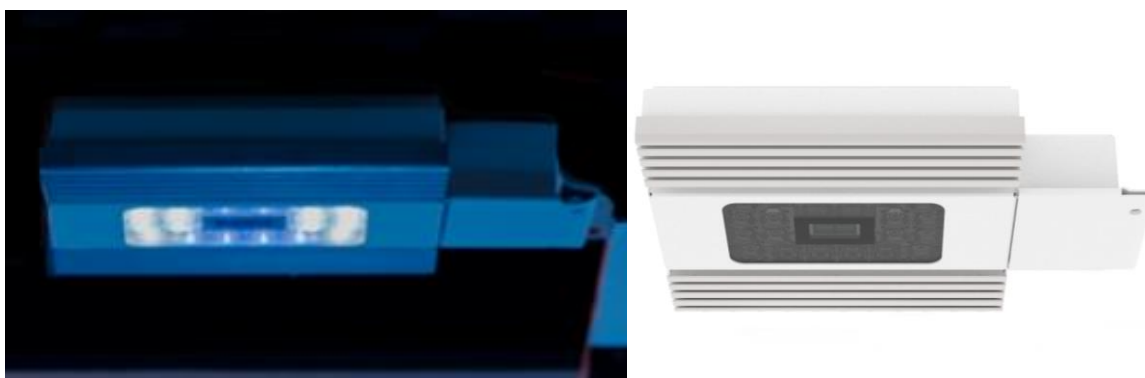


Рис. 1.8 Передавач Li-Fi системи, під'єднаний до стелі. Зліва в увімкненому стані, справа у вимкненому. [36]

Третій елемент – фотодавачі або фотоприймачі. Це сенсори світла або іншої електромагнітної енергії, зображені на Рис. 1.9. Фотодетектори перетворюють світлові сигнали в струм (напругу). [37]

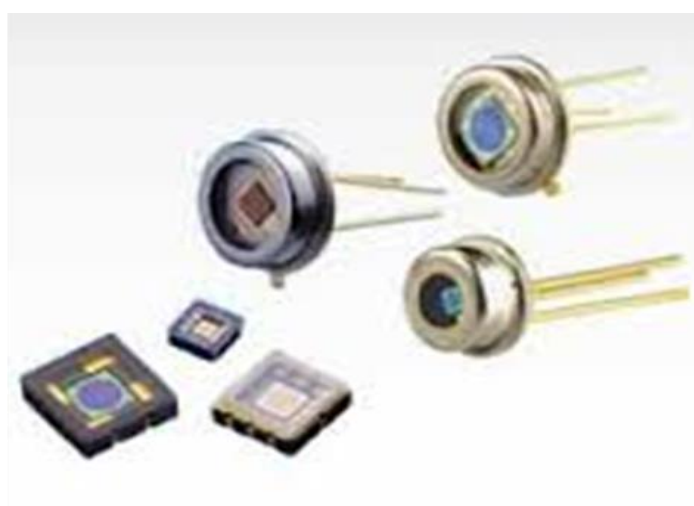


Рис. 1.9 Приклади фотодатчиків і фотоприймачів.

Принцип фотодетектування засновано на двох типах фотоелектричних ефектів. Зовнішній фотоэффект: електрони стають вільними від поверхні металу

внаслідок поглинання енергії, отриманої на потоках потрапляючих фотонів. Наприклад, вакуумний фотодіод побудований на принципі зовнішнього фотоэффекту. Внутрішній фотоэффект: вільні носії заряду генерують шляхом поглинання падаючих фотонів в напівпровідник і р-п перехід фотодіода. [37, 38]

Приклад системи передавання наведено на Рис. 1.10, на якому наведено, що Li-Fi приймач під'єднується через порт USB безпосередньо до кінцевого пристрою. [39]

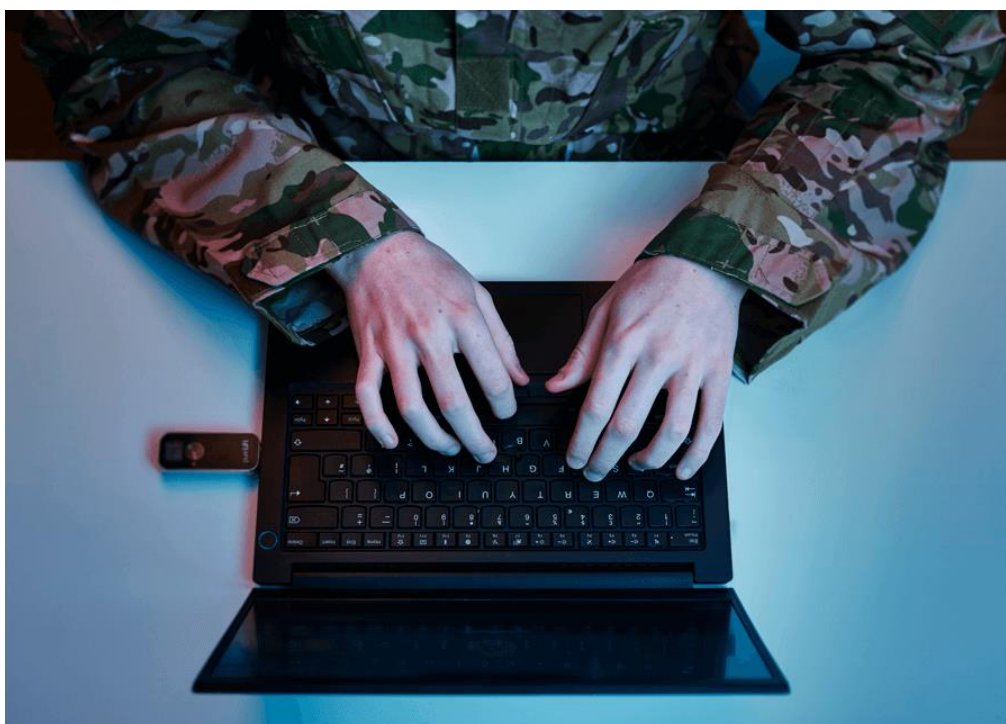


Рис. 1.10 Приклад приймальної сторони Li-Fi системи. [40]

Деякі сучасні приймачі є дуже маленькі та можуть поміститися в корпус смартфона, як зображено на Рис. 1.11.

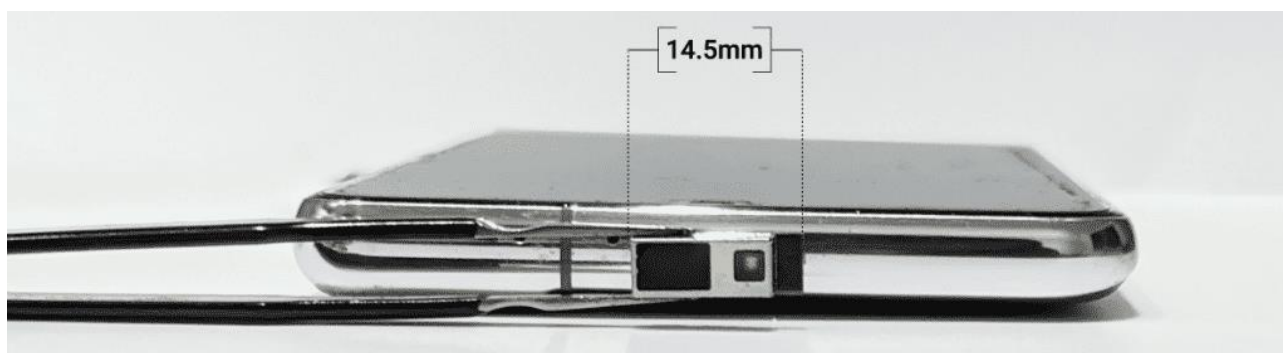


Рис. 1.11 Компактний Li-Fi приймач [41]

VLP алгоритми для приміщень використовують різні показники, такі як потужність прийнятого сигналу (RSS), кут прибуття (AOA), час прибуття (TOA) та різницю в часі прибуття (TDOA). Проте, VLP на основі RSS є особливо популярним в дослідженнях, оскільки він вимагає мало апаратних ресурсів та його реалізація є простою[5]. Серед механік пошук є створення бази даних опорних точок Fingerprint та пошук за ними, сімейство геометричних методів триангуляції та метод Proximity. [42]

1.3 Огляд методів внутрішнього позиціонування

У цьому розділі наведено короткий вступ про застосування, переваги, недоліки, класифікацію і теоретичну основу методів позиціонування всередині приміщень. [43]

Для визначення місцеположення об'єкта потрібно для початку виміряти певну змінну, значення якої залежить від місцеположення в кімнаті. Є три основні змінні та чотири методи: це потужність, кут, та час та чотири методи, TOA (Time of Arrival, час прибуття), TDOA (Time difference of Arrival, різниця часу прибуття), AOA (Angle of Arrival, кут прибуття) або RSS (Received Signal Strength, сила прийнятого сигналу). [44]

1. Received signal strength

Метод визначення за потужністю визначення сигналу (RSS, Received signal strength) є основним методом. Цей метод заснований на вимірюванні відстані за втратою потужності сигналу від передавача до приймача. У ньому вимірюється потужністю прийнятого сигналу. Він має високу точність та не потребує жодного додаткового обладнання, оскільки кожен пристрій сам по собі вже перетворює прийняти сигнал різної потужності, який залежить від потужності прийнятого сигналу. [45]

Сигнали видимого світла послаблюються зі збільшенням відстані між опорним вузлом і передавачем. Де потужність сигналу видимого світла, що приймається контуром приймача і виражається в одиницях дБ м. Деякі методи

локалізації, засновані на RSS, використовують оптичну модель поширення для розрахунку відстані між опорним вузлом і передавачем. [46]

Перевага RSS полягає в наступному: метод RSS може забезпечити високу точність в умовах малих відстаней і LOS, і його можна використовувати в системах без необхідності встановлення нового обладнання як в мережі, так і в цільових вузлах. Таким чином, RSS все ще простіший і дешевший, ніж інші методи, які вимагають встановлення нового обладнання в різних вузлах системи, і він має ще одну суттєву перевагу, яка полягає в тому, що він не потребує синхронізації часу між вузлами системи. [47]

Для вимірювання RSS Вимірювальна модель для Li-Fi, коефіцієнт підсилення каналу по постійному струму в системах Li-Fi пропорційний інтенсивності випромінювання. [48]

2. Angel of Arrival

Метод куту прийому (AOA, Angel of Arrival) базується на куті приймання. Він також дозволяє точно визначити місцеположення об'єкта. Від одного передавача вимірює одразу два кути: азимут ' α ' у площині паралельній площині передавачів та кут падіння θ , у площині паралельній до площини передавачів. Цей метод зображено на Рис. 1.12.

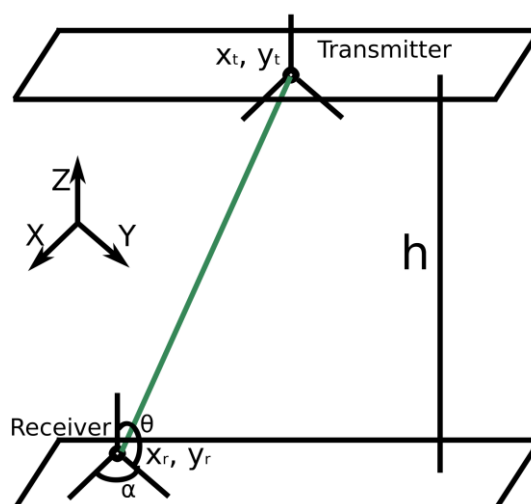


Рис. 1.12 Геометричне зображення методу Angel of Arrival

Вимірявши азимут α , кут приходу θ та знаючи висоту та координати x , y передавача можна визначити місцеположення шуканого об'єкта за формулами (1.1) та (1.2) координати x та y відповідно. [48]

$$x_r = x_t + h \cdot \tan(\theta) \cdot \cos(\alpha) \quad (1.1)$$

$$y_r = y_t + h \cdot \tan(\theta) \cdot \sin(\alpha) \quad (1.2)$$

де x_t , y_t координати передавача

x_r , y_r – координати приймача

θ – кут приходу

α – азимут

h – висота

Перевагою цього методу є в тому, що він одразу вимірює два параметра і тому за однією лампою, якщо відомо висоту шуканого об'єкту, то можна визначити лише за однією лампою його місцеположення. серед недоліків потреба у додатковому обладнанні у вигляді спеціальних матриць фотодетекторів або лінз. а також ще у гіроскопів та акселерометрів, через те що приймач може змінювати свої кути нахилу, через що виникає потреба у компенсації. [49]

3. Time of Arrival

Метод часу приймання (ТОА, Time of Arrival) базується на вимірюванні часу, за який проходить промінь від передавача до приймача. Для знаходження відстані між передавачем та приймачем потрібно помножити швидкість променя, а це швидкість світла, на час, за який це промінь подолав цю відстань, як описано у (1.3) та представлено на Рис. 1.13.

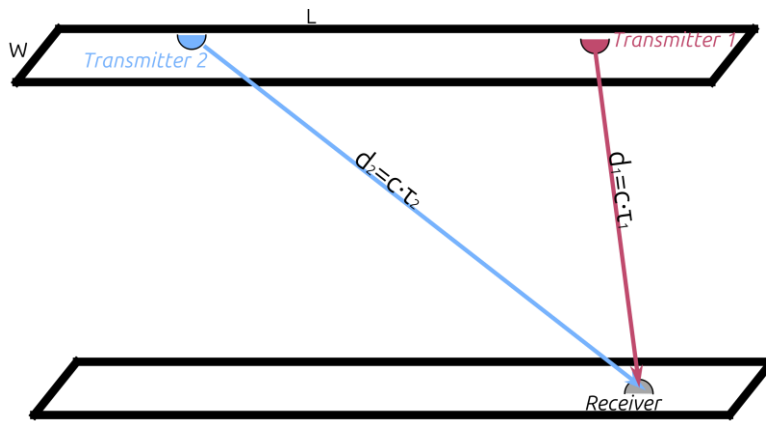


Рис. 1.13 Геометричне зображення методу Time of Arrival
 $(x_r - x_{t.i})^2 + (y_r - y_{t.i})^2 + h^2 = (c \cdot \tau_i)^2$ (1.3)

де

x_r, y_r – координати приймача

$x_{t.i}, y_{t.i}$ – координати i -того передавача

h – висота кімнати

c – швидкість світла

τ_i – час доставки повідомлення між передавачем та приймачем

Недоліком цього методу є те, що промінь проходить цю відстань дуже швидко, і похибка на одну секунду еквівалентна помилці в 30 см. Для цього методу потрібне обладнання вимірювання високоточного часу як на передачі так на приймачі і синхронізація між цими високоточними годинниками. [50]

4. Метод різниці часу приймання (TDOA, Time difference of Arrival) також базується на часі за який промінь проходить передавача до приймача, але замість системі синхронізації використовує кілька передавачів. Він базується на тому, що значення різниці прийому часу між різними передавачами можлива лише в певних точках кімнати, яка утворює гіперболу (1.4). За перетином гіпербол можна визначити точне місцеположення об'єкта в приміщенні. [51]

$$\Delta d = c \cdot (t_1 - t_2) \quad (1.4)$$

Таблиця 1.1

Класифікація методів внутрішнього позиціонування

Назва методу	Зміна, яка вимірюється	Обладнання	Точність
RSS	Потужність (прийнятого сигналу)	Базовий фотоприймач	Висока
AOA	Кут (падіння)		Висока
TOA	Час	Високоточний годинник (на передавачі та приймачі), високоточна система синхронізаціі	Низька
TDOA	Час	Високоточний годинник (на приймачі)	Низька

Наступна методологія стосується способу обробки параметрів. Які саме параметри залежить від методів, розглянутих раніше.

5. Триангуляція

Перше сімейство методів називається триангуляцією. Цей метод базується на розрахунку координат вирішенням системи рівнянь, знаючи кути та/чи відстані між передавачами та приймачами. Для Li-Fi найчастіше використовується трилятераця, тобто визначення координат об'єкта використовуючи відстані між передавачами та приймачем. Вона представлена у (1.5) Цей метод не потребує значного використання пам'яті, обчислювальних потужностей, але нестійкий до неточних вихідних значень. [52]

$$(x_r - x_{t,i})^2 + (y_r - y_{t,i})^2 + h^2 = (d_i)^2 \quad (1.5)$$

x_r, y_r – координати приймача

$x_{t,i}, y_{t,i}$ – координати i -того передавача

h – висота кімнати

d_i – відстань між передавачем та приймачем

6. Fingerprint

Наступний метод Fingerprint. Цей метод працює у два етапи. Перший етап це створення бази даних. Для її створення потрібно визначити точки, значення від яких буде записано в базу даних. Для оптимізації результатів точки знаходяться на певній відстані одна від одної, створюючи сітку, як зображено на Рис. 1.14 [53]

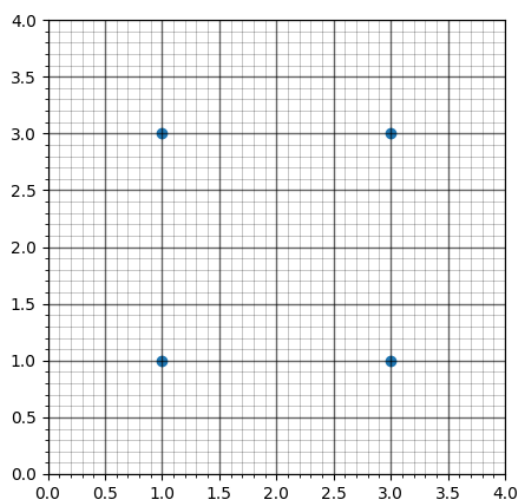


Рис 1.14 Шаблон для створення бази даних Fingerprint із точністю 10 см.

Чим більша кількість опорних точок, тобто чим менша відстань між опорними точками, тим вища точність, але тим більше потрібно зберігати значень у базі даних. Змінюючи ці показники можна досягти потрібного балансу точності визначення місцеположення об'єкта та використаних потужностей. Для кожної опорної точки відповідає стільки значень, скільки розміщено ламп на стелі. Спочатку визначається точність, із якою буде проведено експеримент. За допомогою вимірювань або симуляції в багатьох точках вимірюється прийняті значення та записуються в базу даних із вихідними координатами. Після створення бази даних наступним кроком вимірюється прийнята потужність об'єкта. Шуканий об'єкт надсилає прийняті значення до Li-Fi контролера, на якому значення порівнюються із значеннями у базі даних. Показником порівняння є значення середньоквадратичного відхилення за усіма лампами в даній точці. Значення із найменшим показником

середнього відхилення вважаємо таким, в якому знаходиться користувач. із значеннями в базі даних. Найбільш схоже значення приймаємо за координати об'єкта. [54]

7. Proximity

Наступний метод Proximity. У ньому приймаються значення координат об'єкта за значення лампи, до якого під'єднаний.

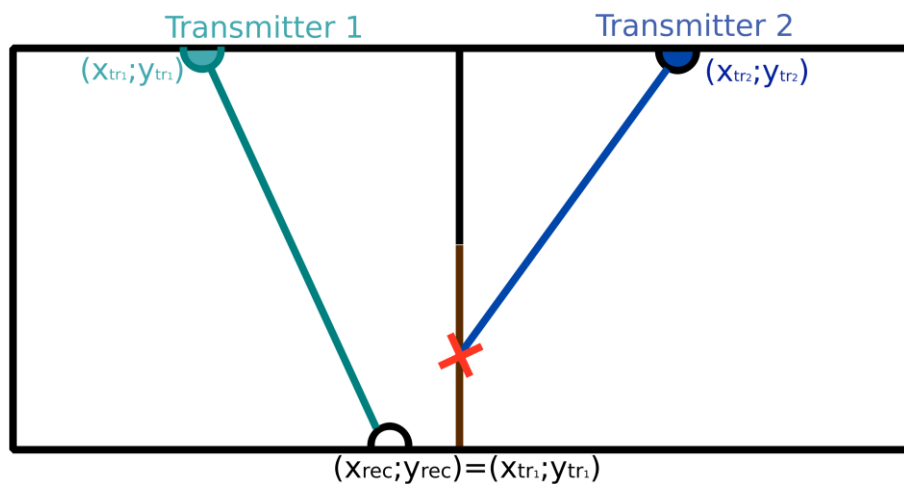


Рис. 1.15 Модель методу Proximity

На Рис. 1.15 користувач отримує сигнал лише від першого передавача, тому він приймає значення координат від нього. Цим методом можна визначити точно, в якій саме кімнаті знаходиться користувач. [55]

Найменш точний та непридатний для знаходження місцеположення об'єкта в кімнаті.

Таблиця 1.2

Класифікація методів внутрішнього позиціонування

Назва методу	Точність	Завадостійкість	Складність
Triangulation	Висока	Низька	Висока
Fingerprint	Достатньо висока	Середня	Середня
Proximity	Дуже низька	Дуже висока	Дуже проста

Висновки

Отже, Li-Fi – це нова сучасна технологія, за якою майбутнє. Частина спектру цієї технології – видиме світло є у 10'000 разів більше за радіодіапазон, тому потенційно вона є набагато швидшою за Wi-Fi та не створює завади радіохвилям, що дозволяє використовувати в різних місцях, де є обмежене використання радіохвиль, такі як лікарні, літаки, реактори, тощо. Також при використанні у внутрішньому позиціонуванні, Li-Fi є набагато точнішою за традиційні радіометоди. Освітлення приміщень, це вже частина готової інфраструктури, що значно зменшує вартість оснащення. Ця технологія має вищу безпеку даних, оскільки світлові промені не виходять із приміщень та їх неможливо перехопити. Але ця технологія залишається новою, обладнання ще досить дороге завади у видимому світлі залишаються проблемою.

Актуальність вирішення проблеми внутрішнього позиціонування набуває збільшує потреби і збільшення кількості інтернет речей, збільшення автоматизації тощо. І є дуже затребувані технології в сучасності.

Завади можуть утворюватися від Сонця, при віддзеркаленні, при несправності передавачів, після проходження чи віддзеркалення від різних поверхностей, але це не досліджується в інших роботах.

Для вирішення поставленої задачі існують різні методи. Методи першої методології показують, який та як параметр вимірюється, за значенням якого буде визначатися місцеположення об'єкта.

Для визначення місцеположення використовуються різні методи, які можна розділити на дві групи. Перша група стосується параметру сигналу, за яким визначаємо місцеположення об'єкта. Перший метод називається RSS та визначається за потужністю сигналу на прийомі. Він простий, точний, не потребує додаткового обладнання. Другий метод це AoA, і в ньому визначається кут приймання. Для цього методу часто потрібне додаткове обладнання як гіроскопи. Останні два методи базуються на вимірюванні часу та різниці часу передавання сигналу від передавача до приймача. Цей метод потребує

високоточного обладнання у синхронізації та вимірюванні часу. Він є найдорожчим.

Наступні методи стосуються способу обробки параметрів. Перше сімейство методів називається триангуляцією. Для Li-Fi найчастіше використовується трилятераця, тобто визначення координат об'єкта використовуючи відстані між передавачами та приймачем. Цей метод не потребує значного використання пам'яті, обчислювальних потужностей, але нестійкий до неточних вихідних значень. Наступний метод Fingerprint. Цей метод працює у два етапи. Перший етап це створення бази даних: за допомогою вимірювань або симуляції в багатьох точках вимірюється прийняті значення та записуються в базу даних із вихідними координатами. Після створення бази даних наступним кроком вимірюється прийнята потужність об'єкта, порівнюється із значеннями в базі даних. Найбільш схоже значення приймаємо за координати об'єкта. Цей метод потребує додаткове місце для зберігання та обчислень, але стійкіший до неправильних початкових значень. Наступний метод Proximity. У ньому приймаються значення координат об'єкта за значення лампи, до якого під'єднаний. Найменш точний

РОЗДІЛ 2

РОЗРОБКА МОДЕЛІ ТА МЕТОДУ ВИРІШЕННЯ ЗАДАЧІ ВНУТРІШНЬОГО ПОЗИЦІОНУВАННЯ ПРИ НАЯВНОСТІ ЗАВАД

2.1 Аналіз методів

У кінці минулого розділу розписано існуючі методи для вирішення задачі внутрішнього позиціонування і з використанням технології Li-Fi.

Для вирішення поставленої задачі обрано метод *fingerprint*, оскільки при *proximity* в нас недостатньо даних для точного позиціонування. Той метод взагалі є не точним поставлених початкових умов. Метод триангуляції потребує складного математичного апарату, але й він не є точним при вирішенні поставленої задачі із завадами. Оскільки при триангуляції, ангуляції та літерації в обчислення входить усі лампи і при необхідності виключити їх із розрахунку при спотворенні на одній з яких, значення дуже сильно відрізняються від інших і складно його змінити. В методі з опорними точками дуже легко виключити спотворені значення з аналізу і аналізувати лише неспотворені значення. І тому в даній роботі обрано метод з опорними точками. При необхідності можна також включити їх для обчислення додаткових значень, коли неможливо точно визначити місцеположення через занадто малу кількість значень, тобто прийнято сигнал від занадто багатьох ламп із спотворенням. [57]

Як показник, на основі якого аналізуємо обрано потужність прийнятої сигналу (RSS). Цей показник підходить найкраще, бо він достатньо легко вимірюється і не потребує додаткового обладнання, на відмінно від інших методів. [58]

В метод із основним показником кута прийняття (AoA) потрібне додаткове обладнання для визначення кута падіння на фотоприймач шуканого об'єкта. Оскільки будь-який нахил самого приймача шуканого об'єкта вносить похибку, то потрібно обладнати гіроскопами та акселерометрами, які

допомагають компенсувати нахил пристрою, щоб точно визначити дійсний кут падіння світла до паралельної площини до площини світлодіодних випромінювачів. [59]

Для методів TDOA та TOA потрібне додаткове обладнання для надточного визначення часу прибуття променів. Для методу TOA також вимагає надточної синхронізації годинників між передавачем і приймачем. Похибка в 1 наносекунду дає помилку в 30 сантиметрів. [60]

Переваги методів AoA, TDOA, TOA над RSS не перевершують затрат на додаткове обладнання системи, яке значно здорожчує систему. Ці методи також містять інші недоліки стосовно RSS.

2.2 Опис структури мережі Li-Fi та Li-Fi моделі

Двонапрямлена система Li-Fi складається з «нижньої», приймальної частини (обладнання користувача) і «верхньої», передавальної частини (обладнання адміністратора мережі). На Рис. 2.1 передавальна частина розташована на стелі (верхній паралелограм), а приймальна на підлозі (нижній паралелограм). Передавальна частина системи Li-Fi складається з LED ламп, приймача оберненого зв'язку, Lamp driver, та Li-Fi контролер, який є центром мережі. Lamp driver керує передаванням перемиканням LED ламп. [61]

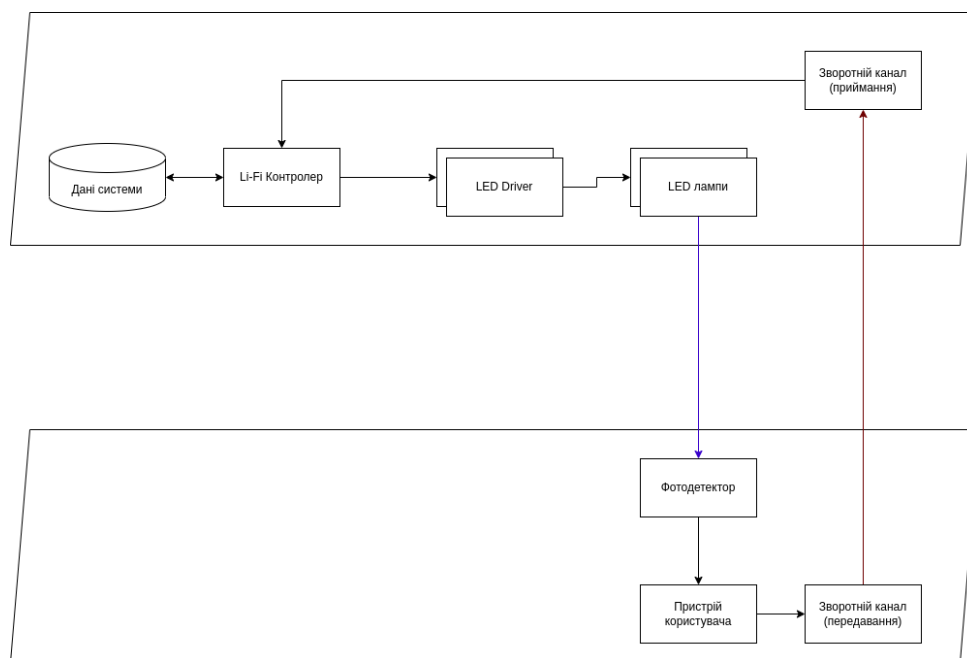


Рис. 2.1 Схема системи Li-Fi

Приймальна частина складається з кінцевого пристрою, до якого під'єднані (або інтегровані) фотоприймач та передавач зворотного зв'язку, а також встановлені драйвера, завдяки яким кінцевий пристрій передає та приймає данні[62, 30-31].

Світлодіодна (LED) лампа призначена для передавання інформації шляхом кодування 1 та 0 оптичним каналом. Lamp driver керує вмиканням та вимиканням LED ламп. Li-Fi контролер є мозком системи і обробляє всю потрібну інформацію, надсилає потрібну інформацію на передавання до Ламп драйверів, обробляє інформацію надіслану користувачем зворотнім каналом та зв'язується з базою даних. База даних зберігає інформацію про систему та керується Li-Fi контролером. Приймання зворотнім зв'язком може бути різним, як світлом чи ІЧ Li-Fi, радіохвилі як Wi-Fi, дротовий зв'язок тощо. Передавальна сторона аналогічно, може бути різноманітною, але у половинні користувача. Фотоприймач приймає світловий сигнал від світлодіодних ламп, а також здатен розрізняти прийняту потужність світлового сигналу. Далі інформація обробляється пристроєм користувача, який також дає команду зворотному каналу на передавання. З'єднання в Li-Fi системі ініціалізує

користувач, надсилаючи запит на з'єднання до Li-Fi, далі передавач відповідає і починається налаштування з'єднання. Передавання даних здійснюється так: Спочатку вони поступають на LED Driver, далі LED Driver керує вмиканням та вимиканням відповідних LED Lamp для передавання даних користувачу. Відсутність сигналу (0) та наявність сигналу (1) відповідають певним значенням потужності випромінювання LED лампах. Далі сигнал на кінцевому пристрої приймається фотоприймачем, підсилюється і перетворюється на електричний струм, звичний для роботи. Передавання сигналу здійснюється подібним чином. Дані поступають на приймач користувача, передаються до приймача зворотного зв'язку. Схема системи Li-Fi для позиціонування користувача зображено на Рис. 2.1. [63, 64]

На Li-Fi контролер зберігається конфігурація системи: Розташування та характеристики LED ламп (потужність випромінювання P , половинний кут випромінювання $\theta_{1/2}$, коефіцієнт посилення оптичного концентратора), розміри кімнати, програма 1 розрахунку потужності в центрі кожного квадрату кімнати розміром $a \times a$. Її варто запускати після зміни конфігурації в кімнаті (LED ламп) та при першому встановленні. При першому приєднанні до Li-Fi системи кінцевий пристрій надсилає свої характеристики (Площа фотоприймача, коефіцієнт перетворення, коефіцієнт заломлення, коефіцієнт посилення оптичного концентратора, кут огляду фотоприймача). [65]

Після налаштування системи, запускаємо програму 1, яка здійснює передобрахунок.

При пошуку розташування пристрою кінцевий користувач надсилає значення прийнятої потужності на Li-Fi контролер. Далі Li-Fi контролер запускає програму 2 в якості вхідних параметрів це отримана потужність на кінцевому пристрої. Результатом буде місце знаходження користувача.

Вони зображені, відповідно, на Рис. 2.2 нижче:

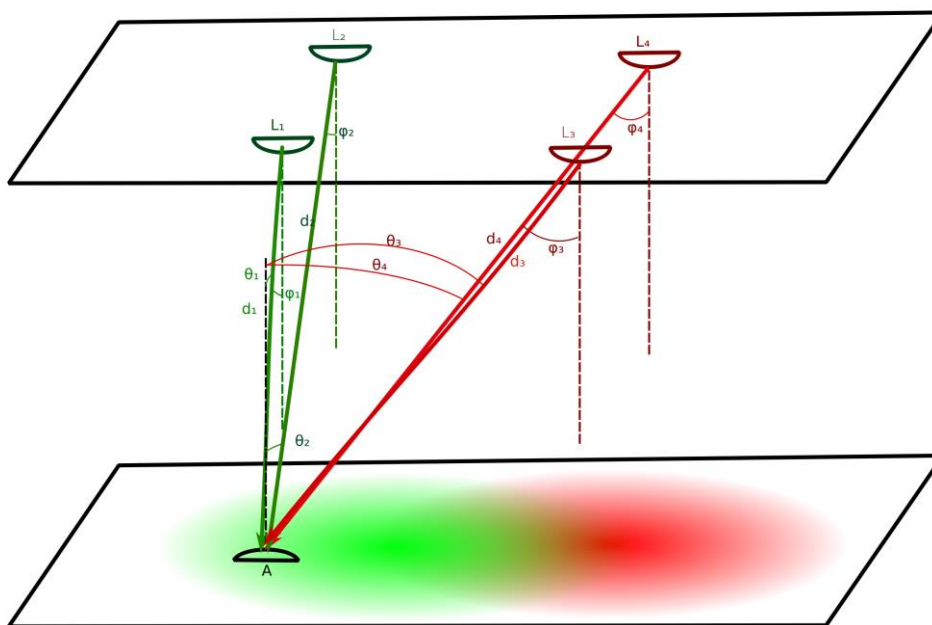


Рис. 2.2 Схеми приміщення із 3-а лампами на стелі та фотоприймачем на підлозі

Як зображено на Рис. 2.2, система Li-Fi для позиціонування містить у кімнаті пристрої кінцевих користувачі на сторонні приймання, LED лампи на передавальній стороні, та світловий канал, по якому передається інформація від передавача (LED лампа) до користувача (фотоприймач). Потужність каналу P_{rec} приймання залежить від наступних змінних[66, 15]:

P_{tr} – потужність передавача;

θ – кут випромінювання;

A – площа покриття фотоприймача;

T – коефіцієнт зміни сигналу від фільтра;

n – показник заломлення;

θ – кут приймача;

L – довжина кімнати;

W – ширина кімнати;

h – висота кімнати, а інші залежать від інших, більш простих параметрів

[15].

І обраховується значення потужності за формулою 2.1:

$$P_{rec} = P_{tr} \frac{(m+1)A}{2\pi d^2} \cos^m(\varphi) Tg(\theta) \cos(\theta) \quad (2.1)$$

Ця формула складна. В ній враховано такі параметри системи:

m – Ламбертовий показник випромінення,

$g(\psi)$ – коефіцієнт посилення оптичного концентратора, складені параметри залежать від простих, та описані у формулах (2.2) - (2.5)

$$m = \frac{\ln(2)}{\ln(\cos\theta)} \quad (2.2);$$

$$g(\theta) = \frac{n^2}{\sin^2(\theta/2)} \quad (2.3);$$

$$d = \sqrt{(x_{tr} - x_{rec})^2 + (y_{tr} - y_{rec})^2 + h^2} \quad (2.4);$$

$$\text{де } x_{rec} \in (0; L); y_{rec} \in (0; W) \quad (2.5).$$

Канал прямої видимості Light of sight (LOS) є основною формулою для знаходження потужності, в Li-Fi системі.

Потужність на фотопримачі рівна значення каналу прямої видимості помножені на потужність LED.

Непрямі промені (NLOS) є відносно малими і складають всього від 10^{-4} до 10^{-2} від променів прямої видимості в залежності від поверхні відбиття. І цей коефіцієнт збільшується із кожним відбиттям експоненційно з показником кількості відбитої поверхні. Тому на даному етапі прийнято рішення знехтувати відбитими променями і тому їх не враховано в кінцевій формулі обрахунку. [67]

Як відомо, для визначення положення користувача в трьохмірному просторі потрібно мінімум три джерела, в системі Li-Fi джерелами є лампи, які розташовані на стелі. В приміщеннях часто розташовано на стелі три або більше ламп. Відповідно зображено на рис. 2.2. [68]

Спочатку розглядається найбільш простий випадок. В даному випадку це 4 лампи розміщені симетрично в кімнаті. Кімната є квадратною в даному випадку.

Вважаємо, що кімнати пусті, а фотоприймач знаходиться на підлозі цієї кімнати, і його висота не змінюються. Тобто є дві площини. Перша площина –

це площина ламп, і друга площина – це площина кінцевого користувача. Між цими площинами висота H , і вони паралельні. [69]

Прийнята потужність обернена пропорційна до квадрату відстані, але вона також залежить від кутів. І оскільки висота в нашій системі не змінюється, але змінюються координати позиціонованого об'єкту на підлозі, і тому змінюються кути приймання та передавання. Залежність потужності від розташування користувача на підлозі є досить складним. Тому раціональним вирішенням поставної задачі, як варіант, є не використання прямого теоретичного обрахунку відстані між лампами та шуканим об'єктом. Як варіант обрано попередній теоретичний обрахунок значень потужності, на підлозі в кімнаті наперед, з наступним пошуком найближчого значення до обрахованого. [70]

На Рис.2.2 представлено модель системи позиціонування на базі мережі Li-Fi та її основні елементи. Передавачі в системі представлені у вигляді світлодіодних LED ламп $\{L_1, L_2, L_3, L_4\}$. Причому лампи, від яких користувач отримує неспотворені сигнали, мають зелені кольори $\{L_1, L_2\}$, а лампи, від яких користувач отримує спотворені сигнали, мають червоні кольори $\{L_3, L_4\}$. Крім того, показані кути передавання і приймання світлових сигналів в мережі Li-Fi. Вони дозволяють розрахувати загасання світлових сигналів в процесі його переміщення від передавача до приймача.[71]

2.3 Методика вирішення задачі визначення місцеположення об'єкта

2.3.1 Складання загального алгоритму

Алгоритм вирішення задачі складається із декількох етапів: 1. Створення бази даних відбитків пальців 2. Визначення які ламп, прийнятий сигнал від яких є спотвореним. 3. Визначення допоміжних точок 4. Визначення координат об'єктів із урахуванням різних сполук ламп (по 2,3,4 ламп). Загальна схема зображена на Рис. 2.3



Рис. 2.3 Загальний алгоритм визначення місцеположення об'єкта в умовах завад

Припустимо, що на лампах L1 та L2 сигнал не спотворений, а на лампах L3 та L4 сигнал спотворений. Також пропустимо, що ми отримати використовуючи інший алгоритм із ламп L1,L2 дві точки. Для визначення яка з цих двох точок є правильною, використаємо інші дві лампи, які є спотвореними. Спочатку візьмемо комбінацію ламп L1,L2,L3.

Знайдемо матрицю різниць потужностей. І для цього порахуємо матрицю різниць Δ для лампи 3.

Беремо координату x_1 та y_1 . Знаходимо різницю потужностей $\Delta(x_1,y_1,L_3)=M(x_1,y_1,L_3)-U(L_3)$. Беремо координати x_2, y_1 та знаходимо $\Delta(x_2,y_1,L_3)=M(x_2,y_1,L_3)-U(L_3)$. Беремо координати x_3, y_1 та знаходимо $\Delta(x_3,y_1,L_3)=M(x_3,y_1,L_3)-U(L_3)$. І так далі збільшуємо координату x . Беремо координати x_{400}, y_1 та знаходимо $\Delta(x_{400},y_1,L_3)=M(x_{400},y_1,L_3)-U(L_3)$. На наступному кроці починаємо знову з x_0 , але замість y_1 беремо y_2 і тоді беремо x_1,y_2,L_3 та обраховуємо $\Delta(x_1,y_2,L_3)=M(x_1,y_2,L_3)-U(L_3)$. Далі беремо x_2,y_2,L_3 та обраховуємо $\Delta(x_2,y_2,L_3)=M(x_2,y_2,L_3)-U(L_3)$. беремо x_{400},y_2,L_3 та обраховуємо $\Delta(x_{400},y_2,L_3)=M(x_{400},y_2,L_3)-U(L_3)$. І знову збільшуємо y на 1, та починаємо x з початку. беремо x_1,y_3,L_3 та обраховуємо $\Delta(x_1,y_3,L_3)=M(x_1,y_3,L_3)-U(L_3)$. І так продовжуємо доки не дійдемо до x_{400},y_{400} : беремо x_{400},y_{400},L_3 та обраховуємо $\Delta(x_{400},y_{400},L_3)=M(x_{400},y_{400},L_3)-U(L_3)$

Створюємо матрицю середньоквадратичних відхилень E розмірністю $x \times y$, отже 400×400 .

$$\text{Обраховуємо } E_{n3}(x_1, y_1) = \sqrt{(\Delta(x_1, y_1, L_1))^2 + (\Delta(x_1, y_1, L_2))^2 + (\Delta(x_1, y_1, L_3))^2}.$$

Оскільки це найперше обраховане значення, то припускаємо що воно найменше $\min_val_{n3} = E_{n3}(x_1, y_1)$, $\min_index_{n3} = (1; 1)$.

Збільшуємо i на одиницю, $i += 1$; Обраховуємо середньоквадратичне відхилення для наступної координати $E_{n3}(x_2, y_1) = \sqrt{(\Delta(x_2, y_1, L_1))^2 + (\Delta(x_2, y_1, L_2))^2 + (\Delta(x_2, y_1, L_3))^2}$.

Порівнюємо $E_{n3}(x_2; y_1) ? \min_val_{n3}$, якщо $E_{n3}(x_2; y_1)$ менше за \min_val_{n3} , тоді $\min_val_{n3} = E_{n3}(x_2; y_1)$ та $\min_index_{n3} = (1; 2)$.

Збільшуємо i на одиницю, $i += 1$; Обраховуємо середньоквадратичне відхилення для наступної координати $E_{n3}(x_3, y_1) = \sqrt{(\Delta(x_3, y_1, L_1))^2 + (\Delta(x_3, y_1, L_2))^2 + (\Delta(x_3, y_1, L_3))^2}$.

Порівнюємо $E_{n3}(x_3; y_1) ? \min_val_{n3}$, якщо $E_{n3}(x_3; y_1)$ менше за \min_val_{n3} , тоді $\min_val_{n3} = E_{n3}(x_3; y_1)$ та $\min_index_{n3} = (1; 3)$.

Продовжуємо збільшувати i до $i = 400$, тобто до x_{400} . Обраховуємо середньоквадратичне відхилення для наступної координати $E_{n3}(x_{400}, y_1) = \sqrt{(\Delta(x_{400}, y_1, L_1))^2 + (\Delta(x_{400}, y_1, L_2))^2 + (\Delta(x_{400}, y_1, L_3))^2}$.

Порівнюємо $E_{n3}(x_{400}; y_1) ? \min_val_{n3}$, якщо $E_{n3}(x_{400}; y_1)$ менше за \min_val_{n3} , тоді $\min_val_{n3} = E_{n3}(x_{400}; y_1)$ та $\min_index_{n3} = (1; 400)$.

Оскільки $x_i = x_{\max}$, тому переходимо до наступної координати y та починаємо з першої координати x . тобто $i = 1; j = 2$, для координат x_1 та y_2 . Обраховуємо середньоквадратичне відхилення для наступної координати $E_{n3}(x_1, y_2) = \sqrt{(\Delta(x_1; y_2; L_1))^2 + (\Delta(x_1; y_2; L_2))^2 + (\Delta(x_1, y_2, L_3))^2}$.

Порівнюємо $E_{n3}(x_1; y_2) ? \min_val_{n3}$, якщо $E_{n3}(x_1; y_2)$ менше за \min_val_{n3} , тоді $\min_val_{n3} = E_{n3}(x_1; y_2)$ та $\min_index_{n3} = (2; 1)$.

Збільшуємо i на одиницю, $i+=1$; Обраховуємо середньоквадратичне відхилення для наступної координати $E_{n3}(x_2,y_2)=\sqrt{(\Delta(x_2;y_2;L_1))^2+(\Delta(x_2;y_2;L_2))^2+(\Delta(x_2;y_2;L_3))^2}$.

Порівнюємо $E_{n3}(x_2;y_2)?\min_val_{n3}$, якщо $E_{n3}(x_2;y_2)$ менше за \min_val_{n3} , тоді $\min_val_{n3}=E_{n3}(x_2;y_2)$ та $\min_index_{n3}=(2;2)$.

Продовжуємо збільшувати i в x_i до того як $x_i=x_{max}$. Обраховуємо середньоквадратичне відхилення для наступної координати $E_{n3}(x_{400},y_2)=\sqrt{(\Delta(x_{400};y_2;L_1))^2+(\Delta(x_{400};y_2;L_2))^2+(\Delta(x_{400};y_2;L_3))^2}$.

Порівнюємо $E_{n3}(x_{400};y_2)?\min_val_{n3}$, якщо $E_{n3}(x_{400};y_2)$ менше за \min_val_{n3} , тоді $\min_val_{n3}=E_{n3}(x_{400};y_2)$ та $\min_index_{n3}=(2;400)$.

Оскільки досягли останнього значення за вісю x , тому збільшуємо значення y та починаємо із першого x . Обраховуємо середньоквадратичне відхилення для наступної координати $E_{n3}(x_1,y_3)=\sqrt{(\Delta(x_1;y_3;L_1))^2+(\Delta(x_1;y_3;L_2))^2+(\Delta(x_1;y_3;L_3))^2}$.

Порівнюємо $E_{n3}(x_1;y_3)?\min_val_{n3}$, якщо $E_{n3}(x_1;y_3)$ менше за \min_val_{n3} , тоді $\min_val_{n3}=E_{n3}(x_1;y_3)$ та $\min_index_{n3}=(3;1)$.

Продовжуємо збільшувати x та y , доки не дійдемо до останньої координати із 160000. Тобто $x_{400};y_{400}$. Тоді Обраховуємо середньоквадратичне відхилення для наступної координати $E_{n3}(x_{400},y_{400})=\sqrt{(\Delta(x_{400};y_{400};L_1))^2+(\Delta(x_{400};y_{400};L_2))^2+(\Delta(x_{400};y_{400};L_3))^2}$.

Порівнюємо $E_{n3}(x_{400};y_{400})?\min_val_{n3}$, якщо $E_{n3}(x_{400};y_{400})$ менше за \min_val_{n3} , тоді $\min_val_{n3}=E_{n3}(x_{400};y_{400})$ та $\min_index_{n3}=(400;400)$.

Знайдено координати точки $p_3=\min_index_{n3}$. Збережемо цю координату та переходимо до пошуку наступної, p_4 по лампах L_1,L_2,L_4 . Для цього знайдемо значення матриці різниць потужностей для цієї матриці.

Беремо координату x_1 та y_1 . Знаходимо різницю потужностей $\Delta(x_1,y_1,L_4)=M(x_1,y_1,L_4)-U(L_4)$. Беремо координати x_2, y_1 та знаходимо $\Delta(x_2,y_1,L_4)=M(x_2,y_1,L_4)-U(L_4)$. Беремо координати x_3, y_1 та знаходимо

$\Delta(x_3, y_1, L_4) = M(x_3, y_1, L_4) - U(L_4)$. І так далі збільшуємо координату x . Беремо координати x_{400}, y_1 та знаходимо $\Delta(x_{400}, y_1, L_4) = M(x_{400}, y_1, L_4) - U(L_4)$. На наступному кроці починаємо знову з x_0 , але замість y_1 беремо y_2 і тоді беремо x_1, y_2, L_4 та обраховуємо $\Delta(x_1, y_2, L_4) = M(x_1, y_2, L_4) - U(L_4)$. Далі беремо x_2, y_2, L_4 та обраховуємо $\Delta(x_2, y_2, L_4) = M(x_2, y_2, L_4) - U(L_4)$. беремо x_{400}, y_2, L_4 та обраховуємо $\Delta(x_{400}, y_2, L_4) = M(x_{400}, y_2, L_4) - U(L_4)$. І знову збільшуємо y на 1, та починаємо x з початку. беремо x_1, y_3, L_4 та обраховуємо $\Delta(x_1, y_3, L_4) = M(x_1, y_3, L_4) - U(L_4)$. І так продовжуємо допоки не дійдемо до x_{400}, y_{400} : беремо x_{400}, y_{400}, L_4 та обраховуємо $\Delta(x_{400}, y_{400}, L_4) = M(x_{400}, y_{400}, L_4) - U(L_4)$

Створюємо матрицю середньоквадратичних відхилень E розмірністю $x \times y$, отже 400×400 .

Обраховуємо $E_{n4}(x_1, y_1) = \sqrt{(\Delta(x_1, y_1, L_1))^2 + (\Delta(x_1, y_1, L_2))^2 + (\Delta(x_1, y_1, L_4))^2}$. Оскільки це найперше обраховане значення, то припускаємо що воно найменше $\min_val_{n4} = E_{n4}(x_1, y_1)$, $\min_index_{n4} = (1; 1)$.

Збільшуємо i на одиницю, $i += 1$; Обраховуємо середньоквадратичне відхилення для наступної координати $E_{n4}(x_2, y_1) = \sqrt{(\Delta(x_2, y_1, L_1))^2 + (\Delta(x_2, y_1, L_2))^2 + (\Delta(x_2, y_1, L_4))^2}$.

Порівнюємо $E_{n4}(x_2; y_1) ? \min_val_{n4}$, якщо $E_{n4}(x_2; y_1)$ менше за \min_val_{n4} , тоді $\min_val_{n4} = E_{n4}(x_2; y_1)$ та $\min_index_{n4} = (1; 2)$.

Збільшуємо i на одиницю, $i += 1$; Обраховуємо середньоквадратичне відхилення для наступної координати $E_{n4}(x_3, y_1) = \sqrt{(\Delta(x_3, y_1, L_1))^2 + (\Delta(x_3, y_1, L_2))^2 + (\Delta(x_3, y_1, L_4))^2}$.

Порівнюємо $E_{n4}(x_3; y_1) ? \min_val_{n4}$, якщо $E_{n4}(x_3; y_1)$ менше за \min_val_{n4} , тоді $\min_val_{n4} = E_{n4}(x_3; y_1)$ та $\min_index_{n4} = (1; 3)$.

Продовжуємо збільшувати i до $i = 400$, тобто до x_{400} . Обраховуємо середньоквадратичне відхилення для наступної координати $E_{n4}(x_{400}, y_1) = \sqrt{(\Delta(x_{400}, y_1, L_1))^2 + (\Delta(x_{400}, y_1, L_2))^2 + (\Delta(x_{400}, y_1, L_4))^2}$.

Порівнюємо $E_{n4}(x_{400};y_1) \geq \min_val_{n4}$, якщо $E_{n4}(x_{400};y_1)$ менше за \min_val_{n4} , тоді $\min_val_{n4}=E_{n4}(x_{400};y_1)$ та $\min_index_{n4}=(1;400)$.

Оскільки $x_i=x_{max}$, тому переходимо до наступної координати y та починаємо з першої координати x . тобто $i=1; j=2$, для координат x_1 та y_2 . Обраховуємо середньоквадратичне відхилення для наступної координати $E_{n4}(x_1,y_2)=\sqrt{(\Delta(x_1;y_2;L_1))^2+(\Delta(x_1;y_2;L_2))^2+(\Delta(x_1;y_2;L_4))^2}$.

Порівнюємо $E_{n4}(x_1;y_2) \geq \min_val_{n4}$, якщо $E_{n4}(x_1;y_2)$ менше за \min_val_{n4} , тоді $\min_val_{n4}=E_{n4}(x_1;y_2)$ та $\min_index_{n4}=(2;1)$.

Збільшуємо i на одиницю, $i+=1$; Обраховуємо середньоквадратичне відхилення для наступної координати $E_{n4}(x_2,y_2)=\sqrt{(\Delta(x_2;y_2;L_1))^2+(\Delta(x_2;y_2;L_2))^2+(\Delta(x_2;y_2;L_4))^2}$.

Порівнюємо $E_{n4}(x_2;y_2) \geq \min_val_{n4}$, якщо $E_{n4}(x_2;y_2)$ менше за \min_val_{n4} , тоді $\min_val_{n4}=E_{n4}(x_2;y_2)$ та $\min_index_{n4}=(2;2)$.

Продовжуємо збільшувати i в x_i до того як $x_i=x_{max}$. Обраховуємо середньоквадратичне відхилення для наступної координати $E_{n4}(x_{400},y_2)=\sqrt{(\Delta(x_{400},y_2;L_1))^2+(\Delta(x_{400},y_2;L_2))^2+(\Delta(x_{400},y_2;L_4))^2}$.

Порівнюємо $E_{n4}(x_{400};y_2) \geq \min_val_{n4}$, якщо $E_{n4}(x_{400};y_2)$ менше за \min_val_{n4} , тоді $\min_val_{n4}=E_{n4}(x_{400};y_2)$ та $\min_index_{n4}=(2;400)$.

Оскільки досягли останнього значення за вісю x , тому збільшуємо значення y та починаємо із першого x . Обраховуємо середньоквадратичне відхилення для наступної координати $E_{n4}(x_1,y_3)=\sqrt{(\Delta(x_1;y_3;L_1))^2+(\Delta(x_1;y_3;L_2))^2+(\Delta(x_1;y_3;L_4))^2}$.

Порівнюємо $E_{n4}(x_1;y_3) \geq \min_val_{n4}$, якщо $E_{n4}(x_1;y_3)$ менше за \min_val_{n4} , тоді $\min_val_{n4}=E_{n4}(x_1;y_3)$ та $\min_index_{n4}=(3;1)$.

Продовжуємо збільшувати x та y , доки не дійдемо до останньої координати із 160000. Тобто $x_{400};y_{400}$. Тоді Обраховуємо середньоквадратичне відхилення для наступної координати $E_{n4}(x_{400},y_{400})=\sqrt{(\Delta(x_{400},y_{400};L_1))^2+(\Delta(x_{400},y_{400};L_2))^2+(\Delta(x_{400},y_{400};L_4))^2}$.

Порівнюємо $E_{n4}(x_{400};y_{400}) \geq \min_val_{n4}$, якщо $E_{n4}(x_{400};y_{400})$ менше за \min_val_{n4} , тоді $\min_val_{n4} = E_{n4}(x_{400};y_{400})$ та $\min_index_{n4} = (400;400)$.

Збережемо $n4 = \min_index_{n4}$. Далі переходимо до обрахунку із використанням 4-х ламп $L_1L_2L_3L_4$. Ця точка буде $n5$.

Створюємо матрицю середньоквадратичних відхилень E розмірністю $x \times y$, отже 400×400 .

Обраховуємо $E_{n5}(x_1,y_1) = \sqrt{(\Delta(x_1,y_1,L_1))^2 + (\Delta(x_1,y_1,L_2))^2 + (\Delta(x_1,y_1,L_3))^2 + (\Delta(x_1,y_1,L_4))^2}$. Оскільки це найперше обраховане значення, то припускаємо що воно найменше $\min_val_{n5} = E_{n5}(x_1,y_1)$, $\min_index_{n5} = (1;1)$.

Збільшуємо i на одиницю, $i += 1$; Обраховуємо середньоквадратичне відхилення для наступної координати $E_{n5}(x_2,y_1) = \sqrt{(\Delta(x_2,y_1,L_1))^2 + (\Delta(x_2,y_1,L_2))^2 + (\Delta(x_2,y_1,L_3))^2 + (\Delta(x_2,y_1,L_4))^2}$.

Порівнюємо $E_{n5}(x_2;y_1) \geq \min_val_{n5}$, якщо $E_{n5}(x_2;y_1)$ менше за \min_val_{n5} , тоді $\min_val_{n5} = E_{n5}(x_2;y_1)$ та $\min_index_{n5} = (1;2)$.

Збільшуємо i на одиницю, $i += 1$; Обраховуємо середньоквадратичне відхилення для наступної координати $E_{n5}(x_3,y_1) = \sqrt{(\Delta(x_3,y_1,L_1))^2 + (\Delta(x_3,y_1,L_2))^2 + (\Delta(x_3,y_1,L_3))^2 + (\Delta(x_3,y_1,L_4))^2}$.

Порівнюємо $E_{n5}(x_3;y_1) \geq \min_val_{n5}$, якщо $E_{n5}(x_3;y_1)$ менше за \min_val_{n5} , тоді $\min_val_{n5} = E_{n5}(x_3;y_1)$ та $\min_index_{n5} = (1;3)$.

Продовжуємо збільшувати i до $i=400$, тобто до x_{400} . Обраховуємо середньоквадратичне відхилення для наступної координати $E_{n5}(x_{400},y_1) = \sqrt{(\Delta(x_{400},y_1,L_1))^2 + (\Delta(x_{400},y_1,L_2))^2 + (\Delta(x_{400},y_1,L_3))^2 + (\Delta(x_{400},y_1,L_4))^2}$.

Порівнюємо $E_{n5}(x_{400};y_1) \geq \min_val_{n5}$, якщо $E_{n5}(x_{400};y_1)$ менше за \min_val_{n5} , тоді $\min_val_{n5} = E_{n5}(x_{400};y_1)$ та $\min_index_{n5} = (1;400)$.

Оскільки $x_i = x_{\max}$, тому переходимо до наступної координати y та починаємо з першої координати x . тобто $i=1; j=2$, для координат x_1 та y_2 . Обраховуємо середньоквадратичне відхилення для наступної координати $E_{n5}(x_1,y_2) = \sqrt{(\Delta(x_1;y_2;L_1))^2 + (\Delta(x_1;y_2;L_2))^2 + (\Delta(x_1;y_2;L_3))^2 + (\Delta(x_1;y_2;L_4))^2}$.

Порівнюємо $E_{n5}(x_1; y_2) ? \min_val_{n5}$, якщо $E_{n5}(x_1; y_2)$ менше за \min_val_{n5} , тоді $\min_val_{n5} = E_{n5}(x_1; y_2)$ та $\min_index_{n5} = (2; 1)$.

Збільшуємо i на одиницю, $i += 1$; Обраховуємо середньоквадратичне відхилення для наступної координати $E_{n5}(x_2; y_2) = \sqrt{(\Delta(x_2; y_2; L_1))^2 + (\Delta(x_2; y_2; L_2))^2 + (\Delta(x_2; y_2; L_3))^2 + (\Delta(x_2; y_2; L_4))^2}$.

Порівнюємо $E_{n5}(x_2; y_2) ? \min_val_{n5}$, якщо $E_{n5}(x_2; y_2)$ менше за \min_val_{n5} , тоді $\min_val_{n5} = E_{n5}(x_2; y_2)$ та $\min_index_{n5} = (2; 2)$.

Продовжуємо збільшувати i в x_i до того як $x_i = x_{\max}$. Обраховуємо середньоквадратичне відхилення для наступної координати $E_{n5}(x_{400}; y_2) = \sqrt{(\Delta(x_{400}; y_2; L_1))^2 + (\Delta(x_{400}; y_2; L_2))^2 + (\Delta(x_{400}; y_2; L_3))^2 + (\Delta(x_{400}; y_2; L_4))^2}$.

Порівнюємо $E_{n5}(x_{400}; y_2) ? \min_val_{n5}$, якщо $E_{n5}(x_{400}; y_2)$ менше за \min_val_{n5} , тоді $\min_val_{n5} = E_{n5}(x_{400}; y_2)$ та $\min_index_{n5} = (2; 400)$.

Оскільки досягли останнього значення за вісю x , тому збільшуємо значення y та починаємо із першого x . Обраховуємо середньоквадратичне відхилення для наступної координати $E_{n5}(x_1; y_3) = \sqrt{(\Delta(x_1; y_3; L_1))^2 + (\Delta(x_1; y_3; L_2))^2 + (\Delta(x_1; y_3; L_3))^2 + (\Delta(x_1; y_3; L_4))^2}$.

Порівнюємо $E_{n5}(x_1; y_3) ? \min_val_{n5}$, якщо $E_{n5}(x_1; y_3)$ менше за \min_val_{n5} , тоді $\min_val_{n5} = E_{n5}(x_1; y_3)$ та $\min_index_{n5} = (3; 1)$.

Продовжуємо збільшувати x та y , допоки не дійдемо до останньої координати із 160000. Тобто $x_{400}; y_{400}$.

Тоді Обраховуємо середньоквадратичне відхилення для наступної координати

$$E_{n5}(x_{400}; y_{400}) = \sqrt{(\Delta(x_{400}; y_{400}; L_1))^2 + (\Delta(x_{400}; y_{400}; L_2))^2 + (\Delta(x_{400}; y_{400}; L_3))^2 + (\Delta(x_{400}; y_{400}; L_4))^2}$$

Порівнюємо $E_{n5}(x_{400}; y_{400}) ? \min_val_{n5}$, якщо $E_{n5}(x_{400}; y_{400})$ менше за \min_val_{n5} , тоді $\min_val_{n5} = E_{n5}(x_{400}; y_{400})$ та $\min_index_{n5} = (400; 400)$.

Збережемо $n5 = \min_index_{n5}$.

Обраховуємо точку $n6 = (n3 + n4 + n5) / 3$, або $x_{n6} = (x_{n3} + x_{n4} + x_{n5}) / 3$ та $y_{n6} = (y_{n3} + y_{n4} + y_{n5}) / 3$. Далі рахуємо відстань $d_{n1n6} = \sqrt{((x_{n1} - x_{n6}))^2 + ((y_{n1} - y_{n6}))^2}$ та

$d_{n_2n_6} = \sqrt{(x_{n_2} - x_{n_6})^2 + (y_{n_2} - y_{n_6})^2}$. Якщо $d_{n_1n_6} < d_{n_2n_6}$, то, вважаємо що правильна точка n_1 а якщо, $d_{n_1n_6} > d_{n_2n_6}$ то вважаємо що правильна точка n_2 . [72]

2.3.2 Алгоритм створення і заповнення бази даних відбитку пальців (F1)

Матриця відбитків пальців зберігає еталону інформацію про прийнятті рівні потужності сигналів об'єктом на підлозі. Місцеположення об'єкта визначається за рахунок порівняння вимірних значень потужності прийняття об'єктом із значеннями в цій базі даних.

Для побудови даної матриці потрібна наступна інформація про світлодіодні лампи, фотоприймач, канал зв'язку, а також параметри приміщення, а саме наступні параметри про приміщення це розміри кімнати, його висота, ширина та довжина. про світлодіодні лампи це їх розташування на стелі, тобто координати по X, Y та z Потужності випромінювання, максимальний кут передавання, коефіцієнт підсилення оптичного концентратора. Про фотоприймач це його максимальний кут приймання, коефіцієнт підсилення оптичного фільтру, показник заломлення. Також для розрахунків важлива точність розрахунку. точність - це відстань між двома сусідніми точками в базі даних, в який проводиться вимірювання. Чим більше ця відстань, тим менше вона займає пам'ять і тим швидше їй дії обробляти, і навпаки чим менше ця відстань, тим більше точність, але дані займають більше місця в пам'яті і тим складніше її обробляти.

Цей алгоритм складається з двох частин. Перша частина - це побудова шаблону для матриці, тобто установка розмірів створення пустих комірок, запит на потрібну інформацію, тощо і друга - це саме її заповнення, тобто обрахунок за формулою значення потужності. обрахунок здійснюється в кожному квадраті на підлозі і кожної квадраті на підлозі відповідає кількість значень рівна кількість ламп від кожної лампи по одному значенню.

Цей алгоритм відбувається у наступні кроки та зображено на Рис. 2.4

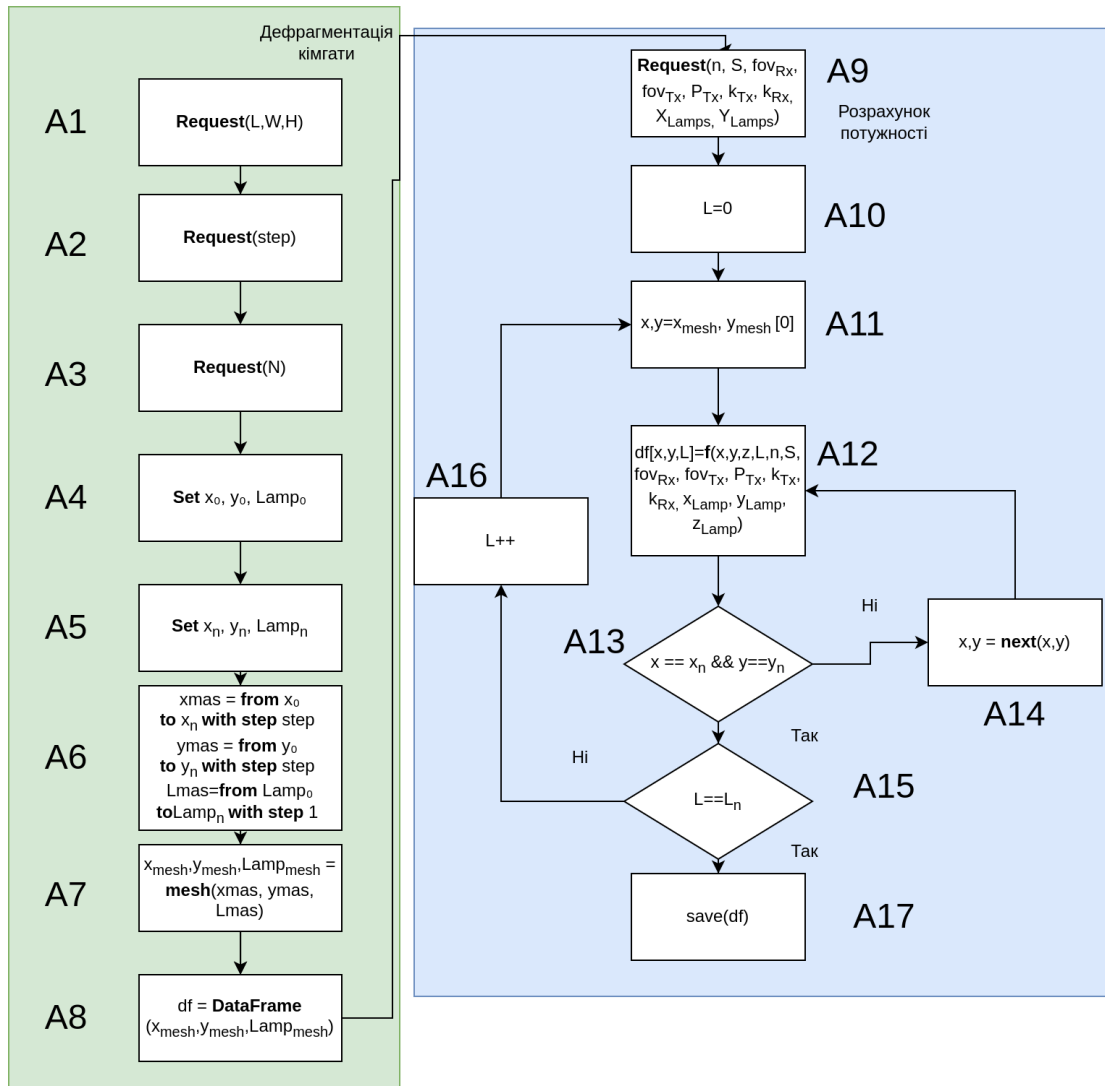


Рис. 2.4. Алогрїт створення бази даних опорних точок Fingerprint

Для початку необхідно здійснити ввід вихідних даних що стосується моделі кімнати, де вирішується задача позиціонування об'єкта приміщення, вводимо вихідні дані щодо моделі кімнати $Q_{\text{мод.кім.}} = f(L, W, H)$

Крок 1. Оператор A1. Вносимо розміри підлоги, а саме її широта та довжина в метрах.

Крок 2 оператор A2 вносимо розмір квадрата, на які ми розбиваємо, тобто ширину та довжину однієї ділянки на підлозі. цей квадрат відповідає значенню в базі даних

Крок 3 Оператор A3 запитуємо про кількість ламп в базі даних

Крок 4 оператора A4 обираємо точку з якою ми починаємо обрахунок для нашої бази даних в тому випадку ми обрали що це координата квадрат, до значення якого також входить координата $x=0$, $y=0$, та найперша лампа, а також відповідає значенню нульових індексів в базі даних

крок 5 оператор A5 точку в даному випадку вона відповідає розміру кімнати а саме це останнє координата по x та по y на підлозі, а також остання лампа.

крок 6 оператор A6 розрахована кількість координат для цього це співвідношення довжини та ширини кімната до сторони 1 квадрат, на які ми ділимо кімнату. записуємо координату кожного квадрату між початкове та кінцевою точкою. кількість квадратів по координаті x дорівнює співвідношенню розмірів кімнати по координаті x , тобто її довжина, до розміру квадрата бічної сторони 1 квадрата, на які ми розбиваємо нашу кімнату. Кількість значень по координаті y складає співвідношенню до ширини кімнати до сторони квадрата, на які розбита кімната.

Крок 8 оператор A8 на даний момент в нас є три масиви X , Y , $Lamps$ розміром $1:n$, $1:m$ і $1:Lamp_size$ за допомогою цих 3 масивів використовуючи функцію `mesh`, яка виконує декартове множення, тобто створюємо усі можливі комбінації з масивів X , Y , $Lamps$, перебираючи усі можливі координати для підлозі для кожної лампи, утворюючи матрицю $m \times n \times Lamp_size$.

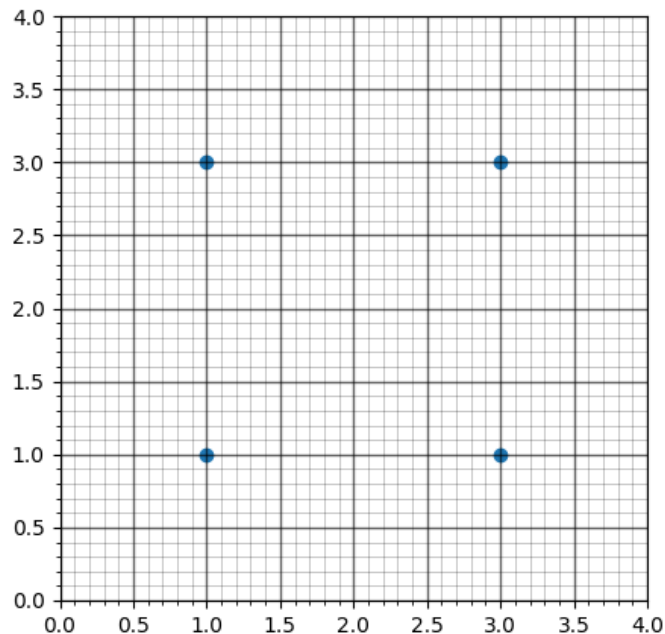


Рис. 2.5 Створення шаблону для бази даних опорних точок Fingerprint

На даному кроці шаблон вже створено, який зображено на Рис. 2.5 і залишилося лише заповнити цю матрицю потрібними значеннями. Минулі оператори A1-A8 не залежать ні від розташування ламп на стелі ні від їх характеристик, лише від характеристик кімнати. З наступного оператора, з A9 по A17 використовуємо окрім інформації про розміри кімнати також інформацію про приймач, передавач, канал зв'язку, а також утворений скелет матриці для заповнення її потрібними значеннями потужності

Крок 9. оператор A9 запитаємо показники передавача та приймача в базі даних а саме координати розташування ламп, їх максимально можливий кут, потужність ламп, підсилення оптичних фільтрів та такі характеристики приймачів як показник заломлення, а також підсилення оптичних концентраторів

Крок 10 Оператор A10 Обираємо найпершу лампу

Крок 11. оператор A11 обираємо початковий квадрат. В даному випадку цей квадратики найближчий до першої координати.

Крок 12. оператор A12 розраховуємо потужність в цьому квадраті

Крок 13 оператор A13 перевіряємо чи це кінцева точка. Якщо це не кінцева точка то переходимо до оператора A14. якщо це кінцева точка то переходимо до оператора A15

Крок 14. оператор A14 зсуваємо на наступну координату і переходимо до оператора A11 тобто розрахунок повторної потужності і входимо знову в цикл

Крок 15. оператор A15 перевіряємо, чи це кінцева лампа, якщо ні, то переходимо до оператора A16. якщо це була остання лампа, то переходимо до оператора A17

Крок 16. оператор A16 зсуваємось на наступну лампу і повертаємось до оператора A10

Крок 17. оператор A17 в нас вже ця Матриця відбитків пальців повністю готова і тому можна її зберігати в базу даних і завершувати цю частину програми

Лістинг програми створення і заповнення бази даних опорних точок:

```
def calculation():
    """
    Створення датафрейму відбитків пальців.
    :return:
    """
    with open('config.yml', 'r') as ymlfile: #відкриття файлу конфігурації
        cfg = yaml.safe_load(ymlfile) #завантаження файлу конфігурацій у змінну cfg

        fov_tx = cfg['fov_tx'] * np.pi / 180 #завантаження граничного кута передавача
        fov_rx = cfg['fov_rx'] * np.pi / 180 #завантаження граничного кута приймача
        n = cfg['n'] #коефіцієнт заломлення
        s = cfg['S'] #площа фотоприймача
        t = cfg['T'] #коефіцієнт підсилення оптичного концентратора
        l = np.array(cfg['L']) #координати x, y, z ламп
        v = np.array(cfg['V']) #розміри кімнати
        step = cfg['step'] #крок точності розрахунку
        sample_rate = cfg['sample_rate'] #
        p_tr = cfg['P_tr'] #потужність передавача
        gfov = (n / np.sin(fov_rx)) ** 2 #коефіцієнт підсилення фільтру
        m = -np.log(2) / np.log(np.cos(fov_tx)) #ламбертовий показник
        lamp_count = l.shape[0] #кількість ламп
        left_shift = step / 2 #зсув зліва (центр першої клітинки)
        right_shift = -step / 2 #зсув справа (центр останньої клітинки)
        xmas = np.arange(left_shift, v[0] + right_shift + step, step).round(3) #масив за координатами x
        ymas = np.arange(left_shift, v[1] + right_shift + step, step).round(3) #масив за координатами y
```

```

df_lamp = pd.DataFrame(1, columns=['x_lamp', 'y_lamp', 'z_lamp']) # створення датафрейму ламп
df_lamp['lamp'] = df_lamp.index #індексація ламп
lamp_mas = np.arange(lamp_count) #Утворення масиву ламп (від 0 до кількості ламп)
x, y, lamps = np.meshgrid(xmas, ymas, lamp_mas) #декартове множення
df_coordinates = pd.DataFrame({'x_fingerprint': x.flatten(), 'y_fingerprint': y.flatten(),
'lamp': lamps.flatten()}) #утворення датафрейму координат
df_coordinates['z_fingerprint'] = 0 #розрахунок для підлоги, z=0 завжди
df = pd.merge(df_lamp, df_coordinates, on='lamp') #об'єднання датафреймів для ламп та для
df['distance'] = np.sqrt((df['x_fingerprint'] - df['x_lamp']) ** 2 + (df['y_fingerprint'] - df['y_lamp']) ** 2 + (
df['z_fingerprint'] - df['z_lamp']) ** 2) #розрахунок відстані для кожної точки на підлозі
df['power_fingerprint'] = (p_tr * s * (m + 1) * t * gfov * (df['z_lamp'] / df['distance']) ** m) / (
2 * np.pi * df['distance'] ** 2) #розрахунок потужності
df['power_fingerprint_int'] = df['power_fingerprint'] / sample_rate #кодування
df['power_fingerprint_int'] = df['power_fingerprint_int'].astype(np.uint16) #переведення в двохбайтне число
df.to_csv('fingerprint_power.csv', index=False) #зберігання

```

2.3.3 Алгоритм визначення не спотворених ламп і потенційних координат місцеположення об'єкта (F2)

Для визначення розташування об'єкта, коли деякі лампи є спотвореними важливо визначити для початку які з ламп є спотворені, а які ні. Для цього розроблено наступного алгоритм, зображений на Рис. 2.6. для визначення нам потрібно створена база даних на минулому кроці а також значення прийнятої потужності від об'єкта, місцеположення якого шукається. Результатом роботи є запис, який містить лампи, які не спотворені, а також потенційні координати, в яких найвірогідніше розташовано об'єкт.

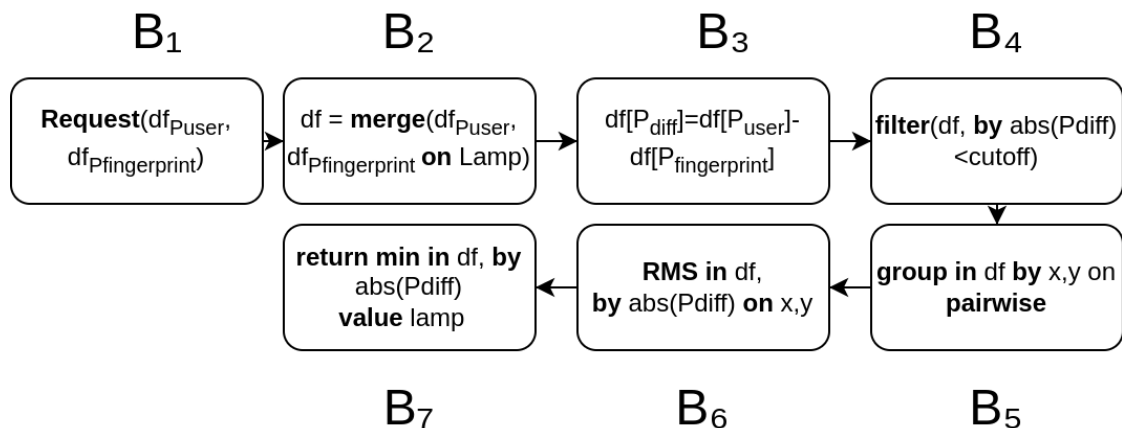


Рис 2.6 Визначення які ламп, прийнятий сигнал від яких є спотвореним.

Крок 1, оператор В1 запитуємо інформацію про потужність користувача та базу даних відбитків пальців

Крок 2 оператор В2 об'єднуємо дата фрейми потужності користувача та еталоні потужності відбитків пальців за спільними лампами

Крок 3 оператор В3 знаходимо різницю в потужності користувача та еталонної потужності записаної в бази даних відбитків пальців

Крок 4 оператор В4 відфільтруємо значення різниці потужності між потужністю користувача та еталонних значень, які є занадто великими залишаючи лише ті значення які відповідають умовам щодо потужності тобто ті, у яких це значення відхилення знаходиться у певних нормах, де вірогідно і є місцеположення користувача

Крок 5 оператор В5 групуємо записи в дата фреймі по координатах x та y попарно, тобто записи про потужність сигналу від різних ламп складаємо поряд.

Крок 6 оператор В6 рахуємо середня квадратичне відхилення по потужності за кожною парою координат в цьому випадку оскільки на минулому кроці згруповано за координатами то в одній координації міститься кілька значень потужності за різними лампами

Крок 7 оператор В7 повертаємо номери ламп в для яких потужність різниця потужності є найменш в усьому датафреймі

На кроці 1 отримуємо наступну інформацію, зображену у таблиці 2.1:

Таблиця 2.1

Частина даних із бази даних опорних точок Fingerprint

Відбитки пальців			
x	y	Lamp	Power
0,05	0,05	1	1,23E-05
0,05	0,05	2	3,45E-05
...
1,75	3,25	2	2,37E-05
1,75	3,25	3	1,95E-05
...
3,95	3,95	3	1,43E-05
3,95	3,95	4	3,15E-05

Також отримано дані про потужність від шуканого об'єкту, зображено у таблиці 2.2.

Таблиця 2.2

Отримані дані вимірної потужності шуканим об'єктом

Потужність користувача	
Lamp	Power
1	2,77E-05
2	2,37E-05
3	1,95E-05
4	1,73E-05

На другому об'єднуємо дані із бази даних опорних точок (Таблиця 2.1) із даними щодо прийнятої потужності об'єктом (Таблиця 2.2) в 1 датафрейм, як зображено у Таблиці 2.3

Таблиця 2.3

Загальний датафрейм по прийнятій потужності

Загальний датафрейм				
x_fingerprint	y_fingerprint	Lamp	Power_fingerprint	Power_user
0,05	0,05	1	1,23E-05	2,77E-05
0,05	0,05	2	3,45E-05	2,37E-05
...	
1,75	3,25	2	2,37E-05	2,37E-05
1,75	3,25	3	1,95E-05	1,95E-05
...	
3,95	3,95	3	1,43E-05	1,95E-05
3,95	3,95	4	3,15E-05	1,73E-05

На кроці 3 рахуємо різницю потужностей $power_difference = Power_user - Power_fingerprint$,. Частину отриманого результату зображено у таблиці 2.4

Таблиця 2.4

Загальний датафрейм по прийнятій потужності із обрахованим значенням різниці потужності

Загальний датафрейм					
x_fingerprint	y_fingerprint	Lamp	Power fingerprint	Power_user	Power_difference
0,05	0,05	1	1,23E-05	2,77E-05	1,54E-05
0,05	0,05	2	3,45E-05	2,37E-05	-1,08E-05
...		
1,75	3,25	2	2,37E-05	2,37E-05	0,00E+00
1,75	3,25	3	1,95E-05	1,95E-05	0,00E+00
...		
3,95	3,95	3	1,43E-05	1,95E-05	5,20E-06
3,95	3,95	4	3,15E-05	1,73E-05	-1,42E-05

На кроці 4 відфільтруємо значення, більші за модулем за 1-Е6. У таблиці 2.5 Зеленим кольором позначено значення, які залишаться після фільтрації, червоним – ні.

Таблиця 2.5

Фільтрація даних у загальному датафреймі по прийнятій потужності

Загальний датафрейм					
x_fingerprint	y_fingerprint	Lamp	Power_fingerprint	Power_user	Power_difference
0,05	0,05	1	1,23E-05	2,77E-05	1,54E-05
0,05	0,05	2	3,45E-05	2,37E-05	-1,08E-05
...		
1,75	3,25	2	2,37E-05	2,37E-05	0,00E+00
1,75	3,25	3	1,95E-05	1,95E-05	0,00E+00
...		
3,95	3,95	3	1,43E-05	1,95E-05	5,20E-06
3,95	3,95	4	3,15E-05	1,73E-05	-1,42E-05

У таблиці 2.6 показано, що більшість значень відфільтровано.

Таблиця 2.6

Відфільтровані значення загального датафрейму

Загальний датафрейм					
x_fingerprint	y_fingerprint	Lamp	Power_fingerprint	Power_user	Power_difference
2,45	3,65	1	0,0000273	2,77E-05	4,00E-07
2,45	3,65	2	0,000024	2,37E-05	-3,00E-07
2,45	3,65	3	0,0000189	1,95E-05	6,00E-07
...	...				0
1,75	3,25	2	2,37E-05	2,37E-05	0,00E+00
1,75	3,25	3	1,95E-05	1,95E-05	0,00E+00

На кроці 5 об'єднуємо залишкові за однаковими координатами x та у певні групи, як зображено у таблиці 2.7

Таблиця 2.7

Об'єднання залишкових значень за спільними координатами у загальному датафреймі

Загальний датафрейм						
index	x_finger print	y_finger print	Lamp	Power_finger print	Power_user	Power_difference
245	2,45	3,65	1	2,73E-05	2,77E-05	4,00E-07
			2	2,40E-05	2,37E-05	-3,00E-07
246	2,45	3,65	1	2,73E-05	2,77E-05	4,00E-07
			3	1,89E-05	1,95E-05	6,00E-07
247	2,45	3,65	2	2,40E-05	2,37E-05	-3,00E-07
			3	1,89E-05	1,95E-05	6,00E-07
...
543	1,75	3,25	2	2,37E-05	2,37E-05	0,00E+00
			3	1,95E-05	1,95E-05	0,00E+00

На кроці 6 рахуємо середньоквадратичне значення різниць потужностей за парами ламп, як зображено у таблиці 2.8.

Таблиця 2.8

Частина із загального датафрейму після обрахунку середньоквадратичного значення потужності за залишковими координатами

Загальний датафрейм			
x_fingerprint	y_fingerprint	Lamp	Power_difference
2,45	3,65	1&2	5,00E-07
2,45	3,65	1&3	7,21E-07
2,45	3,65	2&3	6,71E-07
...
1,75	3,25	2&3	0

На кроці 7 повертаємо кінцеве значення. Як можна побачити, у таблиці 2.8, у рядку, виділеному зеленим, значення 0 мінімальне. Це значення відповідає лампам 2 та 3, отже вони не є спотвореними

Лістинг програми визначення, від яких ламп сигнал прийнято без спотворень

```
def find_location(df_user, df_fingerprint, save_result):
    df_with_error = calculate_error(df_fingerprint, df_user)
    df_lamp_combos, lamp_combo = select_best_suit(df_with_error)

def calculate_error(df_fingerprint, df_user):
    df = pd.merge(df_fingerprint, df_user, on='lamp') #об'єднання датафреймів відбитків пальців та
    сигналів,
    # прийнятих користувачем в новий dataframe за колонкою лампи
    df['diff'] = df['power_fingerprint'] - df['power_user_distorted'] #розрахунок різниці потужності
    return df

def select_best_suit(df):
    cond = np.abs(df['diff']) < CUTOFF #створення умови фільтрування
    df_filtered = df[cond] # фільтрування
    df_intersection = df_filtered[df_filtered.duplicated(subset=['x_fingerprint', 'y_fingerprint'], keep=False)]
    #знаходимо лише ті значення, які зустрічаються щонайменше 2 рази на 1 координату
    df_merged = df_intersection.groupby(['x_fingerprint', 'y_fingerprint']).apply(df_applying,
                                                                                include_groups=False).reset_index()
    #.agg({'diff': root_mean_square, 'lamp': tuple_function}).reset_index()
    #об'єднання у можливі пари
    lamp_combos = df_merged['lamp_combo'].unique() #Усі пари ламп, що наявні в датафреймі
    cut_cond = df_merged['diff_rms'] <= CUTOFF_EXTREME #сильніші межі фільтрування
    for lamp_combo in lamp_combos: #для кожної комбінації ламп
        df_subset = df_merged[df_merged['lamp_combo'] == lamp_combo] #лише цього
        min_diff = df_subset['diff_rms'].min() #пошук мінімального значення
        cut_cond = cut_cond | (df_subset['diff_rms'] <= min_diff * 1.1) #залишаємо значення які більше за
    статистичну похибку АБО близько до мінімального значення
    df_cuttred = df_merged[cut_cond] #фільтруємо
    df_the_best = df_cuttred.loc[df_cuttred.diff_rms.abs() <= CUTOFF_EXTREME,:].copy() #найкращий
    запис в датафреймі
    best_lamp_combo = df_the_best.iloc[0]['lamp_combo'] # виписка ламп із найкращими координатами
```

2.3.4 Визначення допоміжних точок (F3)

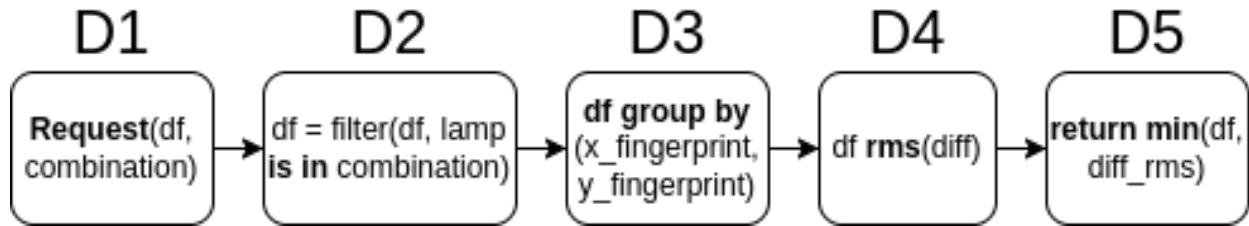


Рис. 2.7 Алгоритм визначення допоміжних точок

Після виконання минулого алгоритму в нас визначено дві точки, в яких найвірогідніше, знаходиться об'єкт. Для визначення в якій саме двох точках знаходиться об'єкт потрібно додаткова інформація. Ця інформація - це допоміжні точки. Ці допоміжні точки визначаються за допомогою алгоритму. Тут визначає потужність, на визначені потужності місцеположення при декількох лампах, включаючи спотворені лампи, і від трьох чи чотирьох ламп. Цей алгоритм зображено на Рис. 2.7

Крок 1 Оператор D1. Запитуємо датафрейм та комбінацію ламп, за якою визначаємо найкращу точку.

Крок 2 Оператор D2. Залишаємо лише ті записи, які належать потрібній комбінації ламп. Відфільтровуємо записи, які відповідають лампам, що не належать потрібній комбінації ламп.

Якщо обрано всі 4 лампи, то ніякі записи на цьому етапі не відфільтровується. Якщо обрано 3 лампи, то відфільтровується лише 1 запис.

Крок 3 Оператор D3. Групуємо записи за координатами. Інакше кажучи, об'єднуємо записи від різних ламп в певній координаті для наступної обробки

Крок 4 Оператор D4. Рахуємо середньоквадратичне значення різниці між потужністю в базі даних відбитків пальців та прийнятою потужністю користувачем за кількома лампами по всім координатам.

Крок 5. Оператор D5. Шукаємо та повертаємо найменше значення в Базі даних.

Лістинг програми визначення допоміжних точок:

```
def find_the_best(df, lamp_combo, auxiliary):
    df_filtered = df[df['lamp'].isin(lamp_combo)] # залишаємо лише ті значення, що відповідають лампам
    aggregated_df = df_filtered.groupby(['x_fingerprint',
    'y_fingerprint'])['diff'].agg(root_mean_square).reset_index()
    # Об'єднуємо за середньоквадратичним значенням
    aggregated_df.rename(columns={'diff': 'diff_rms'}, inplace=True)
    # перейменовуємо колонку, оскільки це вже не 1 значення різниці, а набір середньоквадратичних
    min_val = aggregated_df.iloc[aggregated_df.diff_rms.idxmin()].copy() # Пошук найменшого значення
    min_val['lamp_combo'] = tuple(lamp_combo) # Додаємо комбінацію ламп
    min_val['auxiliary'] = auxiliary # Додаємо значення допоміжності
    return min_val # Повертаємо значення
```

2.3.5 Визначення координати об'єкта серед потенційно можливих точок із використанням допоміжних точок (F4)

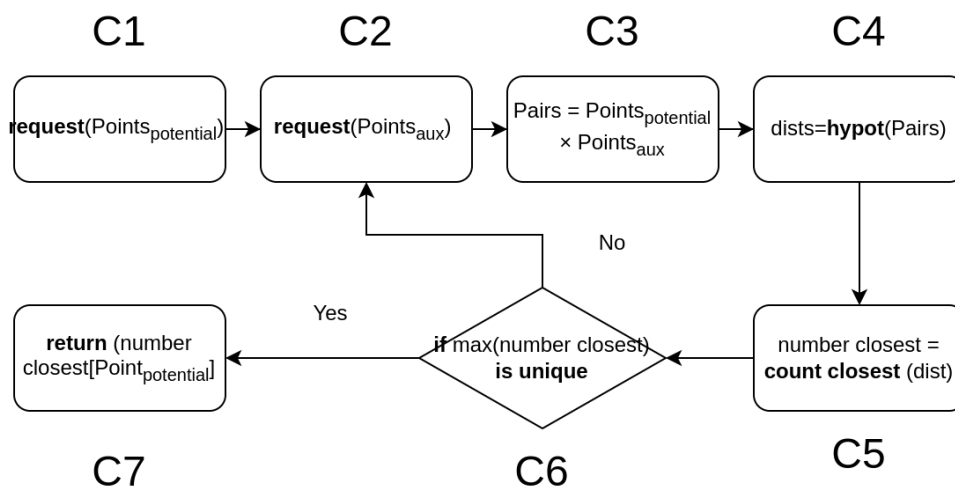


Рис. 2.8 Алгоритм визначення координати об'єкта серед потенційно можливих точок із використанням допоміжних точок

Із використанням допоміжних точок можна визначити в якій із цих двох точок знаходиться об'єкт. В цьому Алгоритмі вимірюються, до якої із двох точок ближче допоміжна точок і чим ближче допоміжних точок до однієї чи іншої потенційної точки і вважаємо за місце розташування об'єкта.

Цей орден працює ітеративно, тобто якщо першій роботі алгоритм не вдалося за допомогою допоміжних точок визначити місцеположення об'єкта серед двох потенційних точок, то операція наступні це трапляється коли дві коли однакова кількість допоміжних точок ближче до однієї та до іншої потенційної точки. Алгоритм зображено на Рис. 2.8.

Запропоновано наступний алгоритм визначення координат об'єкта:

Крок 1. Оператор С1. Запитуємо точки, в яких потенційно може знаходитися об'єкт (одна точка правильна, інша точка помилкова, обидві лампи не спотворені)

Визначаємо $P_{potential1}$; $P_{potential2}$;

Крок 2. Оператор С2. Запитуємо допоміжні точки (запитуємо послідовно, від двох ламп: від однієї неспотвореної та однієї спотвореної лампи,

від сполуки із 3 ламп, серед яких 2 не спотворені і 1 спотворення = 2 точки P_{3aux1} , P_{3aux2} запитуємо допоміжну точку від 4 ламп. = точка P_{4aux1}

Крок 3. Оператор С3. Створюємо усі можливі сполуки потенційних координат та допоміжних координат використовуючи декартове множення.

Визначаємо, яка з точок $P_{potential1}$; $P_{potential2}$; є достовірною (правильною). $P_{доставірна}$

Крок 4. Оператор С4. Рахуємо відстань між потенційними та допоміжними точками.

Розраховуємо відстань між $P_{potential1}$; та P_{3aux1} , P_{3aux2} P_{4aux1}

Відстань між $P_{potential1}$ - P_{3aux1} = $d_{1\ potential1,3aux1}$

Відстань між $P_{potential1}$ - P_{3aux2} = $d_{2\ potential1,3aux2}$

Відстань між $P_{potential1}$ - P_{4aux1} = $d_{3\ potential1,4aux1}$

Розраховуємо відстань між $P_{potential2}$; та P_{3aux1} , P_{3aux2} P_{4aux1}

Відстань між $P_{potential2}$ - P_{3aux1} = $d_{10\ potential2,3aux1}$

Відстань між $P_{potential2}$ - P_{3aux2} = $d_{20\ potential2,3aux2}$

Відстань між $P_{potential2}$ - P_{4aux1} = $d_{30\ potential2,4aux1}$

Крок 5. Оператор С5. Рахуємо кількість допоміжних точок які ближче до кожної потенційної точки.

Визначаємо мінімальне значення відстаней серед пар відстаней між d_1 та d_{10} , d_2 та d_{20} , d_3 та d_{30} . Які допоміжні точки до якої лампи ближче.

Визначаємо, до якої потенційної точки ближче допоміжні точки. Для цього порівнюємо відстані d_1 та d_{10} , d_2 та d_{20} , d_3 та d_{30} . $d_1 - d_{10} = \Delta_{1,10}$. Якщо $\Delta_{1,10} < 0$, то P_{aux1} ближче до $P_{potential1}$ ніж $P_{potential2}$

Крок 6. Оператор С6. Перевіряємо, чи лише до однієї потенційної точки більшість допоміжних точок ближчі. Якщо ні, то обираємо інші допоміжні точки і переходимо до Оператору С2. Якщо так, то переходимо до оператора С7

Крок 7. Оператор С7. Вважаємо ту потенційну точку, до якої найбільша кількість допоміжних точок ближче – правильною.

Лістинг програми розрахунок відстані між потенційними та допоміжними точками:

```
def calculate_distances_with_opposite_flags(df, auxilarity):
    """
    Calculates the distances between points with opposite flags in a DataFrame.

    Args:
        df: A pandas DataFrame with columns 'x', 'y', and 'auxiliary'.

    Returns:
        A new pandas DataFrame with columns 'x1', 'y1', 'x2', 'y2', 'distance'.
    """

    # if not all(col in df.columns for col in ['x', 'y', 'flag']):
    #     raise ValueError("DataFrame must contain 'x', 'y', and 'flag' columns.")

    # Separate points based on their flag
```

```

flag1_points = df[df['auxiliary'] == 0][['x_fingerprint', 'y_fingerprint']].values
flag2_points = df[df['auxiliary'].isin(auxilarity)][['x_fingerprint', 'y_fingerprint']].values

flag1_df = df[df['auxiliary'] == 0][['x_fingerprint', 'y_fingerprint']]
flag2_df = df[df['auxiliary'].isin(auxilarity)][['x_fingerprint', 'y_fingerprint']]

# Create a list of tuples containing the point coordinates and distances
results = []

if len(flag2_points) == 0:
    if len(flag1_points) == 1:
        results.append({
            'x1_fingerprint': flag1_df.iloc[0]['x_fingerprint'],
            'y1_fingerprint': flag1_df.iloc[0]['y_fingerprint'],
            'x2_fingerprint': flag1_df.iloc[0]['x_fingerprint'],
            'y2_fingerprint': flag1_df.iloc[0]['y_fingerprint'],
            'distance': 0
        })
        result_df = pd.DataFrame(results)
        return result_df
    else:
        result_df = pd.DataFrame(results)
        return result_df

# distances = 0
# if len(flag2_points) == 1:
#     flag2_points = np.array([flag2_points[0], flag2_points[0]])
flag1_points = np.array(flag1_points, dtype=float)
flag2_points = np.array(flag2_points, dtype=float)
# distances = hypot_array_one_value(flag1_points, flag2_points[0])

# Calculate distances using cdist
distances = cdist(flag1_points, flag2_points)

```

```

for i in range(distances.shape[0]):
    for j in range(distances.shape[1]):
        results.append({
            'x1_fingerprint': flag1_df.iloc[i]['x_fingerprint'],
            'y1_fingerprint': flag1_df.iloc[i]['y_fingerprint'],
            'x2_fingerprint': flag2_df.iloc[j]['x_fingerprint'],
            'y2_fingerprint': flag2_df.iloc[j]['y_fingerprint'],
            'distance': distances[i, j]
        })

# Create a DataFrame from the results
result_df = pd.DataFrame(results)

return result_df

```

Пошук найменшої відстані

```

def find_minimal_distance(df):
    if df.empty:
        return NO_COORDINATES

    df_grouped = df.groupby(['x2_fingerprint', 'y2_fingerprint'])
    df_minimal_value = df[df['distance'] <= df_grouped['distance'].transform('min') * 1.001]
    df_counted = df_minimal_value.value_counts(subset=['x1_fingerprint',
                                                         'y1_fingerprint']) #df_minimal_value.groupby(['x1_fingerprint',
                                                         'y1_fingerprint']).size()
    max_value = df_counted.max()
    is_max_unique = (df_counted == max_value).sum() == 1

    if is_max_unique:
        return df_counted.idxmax()

    else:
        return NO_COORDINATES

```

Поставлена задача вирішується у два етапи. Перший це визначення, від яких ламп сигнал є спотворений а другий це за допомогою цієї інформації визначення місцеположення об'єкта.

Висновки

Для вирішення поставленої задачі визначення місцеположення об'єкта обрано методи RSS, який у якості вимірювального параметру використовує потужністю прийнятого сигналу приймачем, та методом Fingerprint, що ґрунтується на попередньому формуванні бази даних опорних точок із подальшим порівнянням поточних параметрів сигналу з еталонними значеннями.

Переходячи до апаратної реалізації, основним компонентом Li-Fi системи є Li-Fi контролер із інтегрованим модулем пам'яті де зберігається у тому числі база даних. Li-Fi контролер керує LED драйверами, які вмикають та вимикають LED лампи, Оптичний сигнал, прийнятий фотодетектором, проходить етап первинної обробки й за необхідності передається зворотним каналом зв'язку до контролера.

Апаратна складова побудована таким чином, що у системі вже є усі необхідні елементи для обчислення місцеположення об'єкта методами RSS та Fingerprint та зберігання бази даних опорних точок.

У межах дослідження розроблено алгоритм локалізації, який передбачає попереднє виділення пам'яті під базу даних відповідно до заданої точності позиціювання та проведення симуляції розподілу потужності в кожній опорній точці приміщення.

Розроблено алгоритм, який вирішує поставлено задачу визначення місцезнаходження об'єктів всередині приміщення за допомогою технології Li-Fi. Перед початком симуляції за попередньою визначеною точністю виділяється місце у пам'яті Li-Fi контролера для бази даних, потім проводиться симуляція розташування в кожній опорній точці, результати якої зберігаються в базу даних для подальших розрахунків. На наступному кроці визначається, сигнал від яких ламп прийнято без спотворень. Для цього визначається середньоквадратичне значення помилки від якої з пари ламп найбільше в порівнянні з іншими. Використовується якщо в нас є лише дві лампи, сигнал від яких прийнято без

спотворень і нам потрібно визначити місцеположення, використовуючи цю пару ламп. Але в частинах випадків є два значення можливого місцеположення шуканого об'єкту, які можуть розташуватися на значній відстані одна від одної. Для вирішення поставленої проблеми розроблено алгоритм визначення допоміжних точок за допомогою яких визначаємо, до якої із цих двох точок ближче ту точку і вважаємо за місцеположення шуканого об'єкту.

РОЗДІЛ 3

РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ ВНУТРІШНЬОГО ПОЗИЦІОНУВАННЯ ПРИ НАЯВНОСТІ ЗАВАД

3.1 Створення бази даних

Результатами обрахунку при першому випадку є матриця, рядок та стовпець якої відповідає значенню прийнятої потужності на підлозі. Матриця велика, безпосереднє числове значення, зручне для огляду людиною неможливе, тому для її представлення обрано кольорову діаграму, у якій колір відповідає певному значенню. Результати представлено на Рис. 3.1.

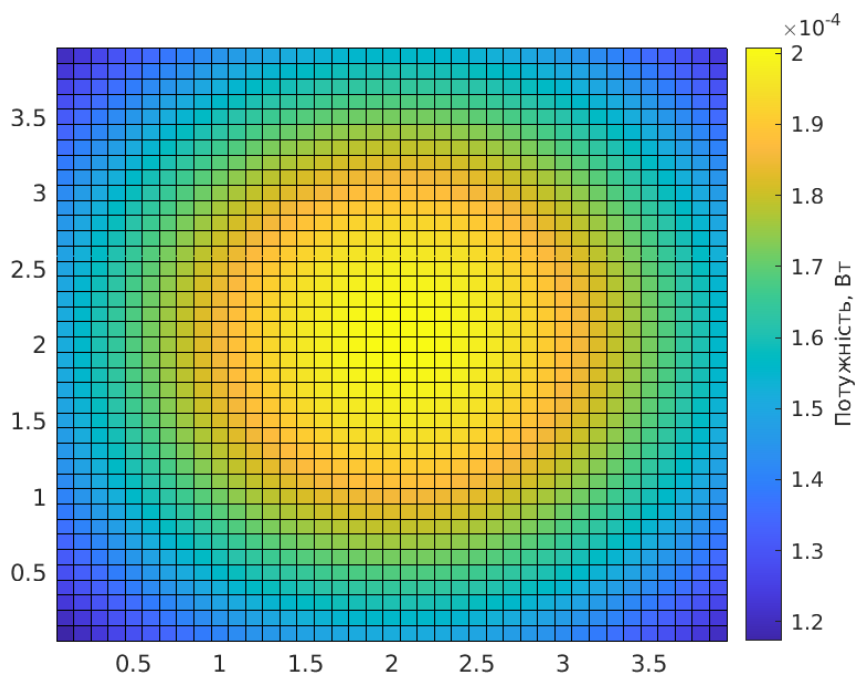


Рис 3.1 Представлення у вигляді кольорової діаграми матриці відбитку пальців при розміщенні однієї лампи на стелі

При використанні кількох ламп для зберігання даних про потужність прийняття сигналу двомірної матриці недостатньо. Кожна двомірна матриця потужності прийнятого сигналу є одним шаром трьохмірної матриці, у якого номер лампи відповідає третій координаті у трьохмірній матриці. Для зображення цієї матриці розділяємо на кожен шар окремо за потужністю прийнятого сигналу від кожної із ламп. Відповідні значення потужності зображенні на Рис. 3.2-3.5

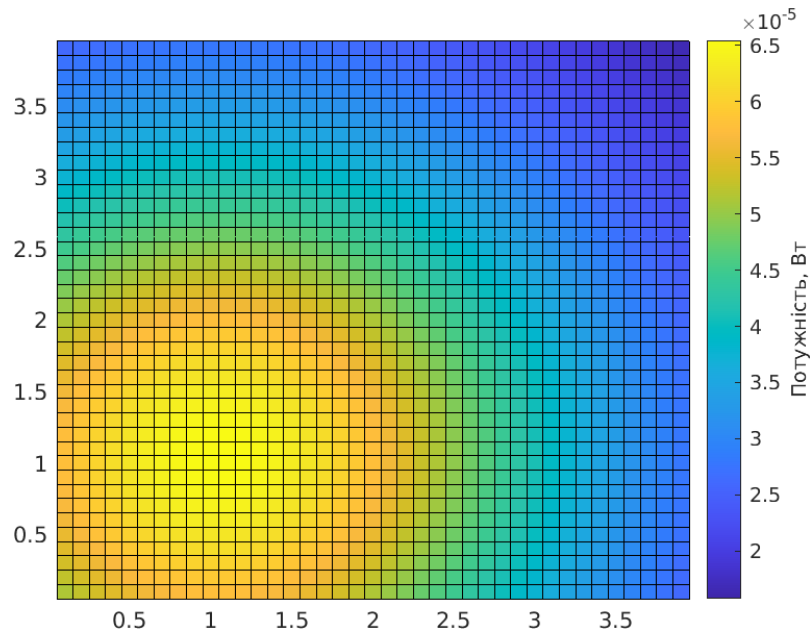


Рис. 3.2 Кольорова мапа потужності прийняття сигналу без спотворень від лампи L_1

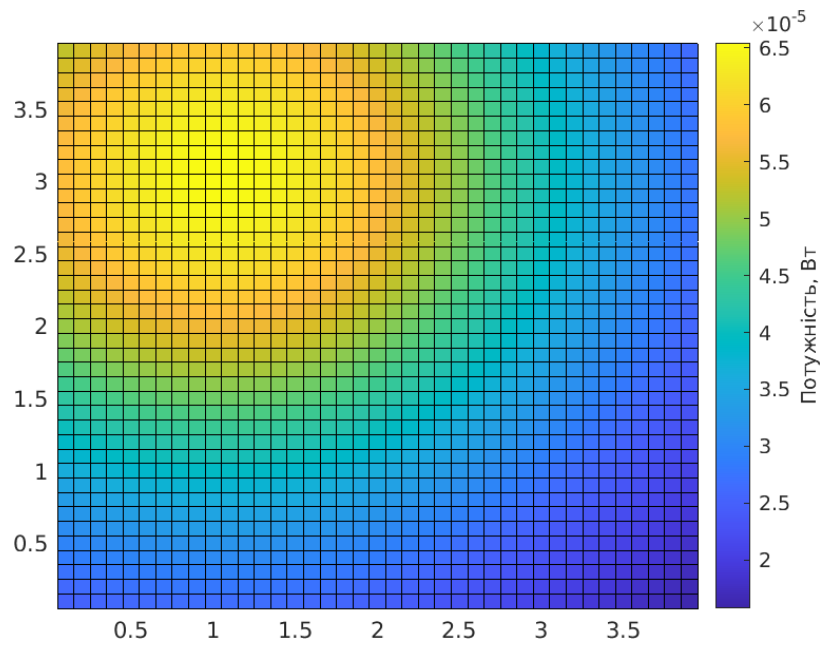


Рис. 3.3 Кольорова мапа потужності прийняття сигналу без спотворень від лампи L_2

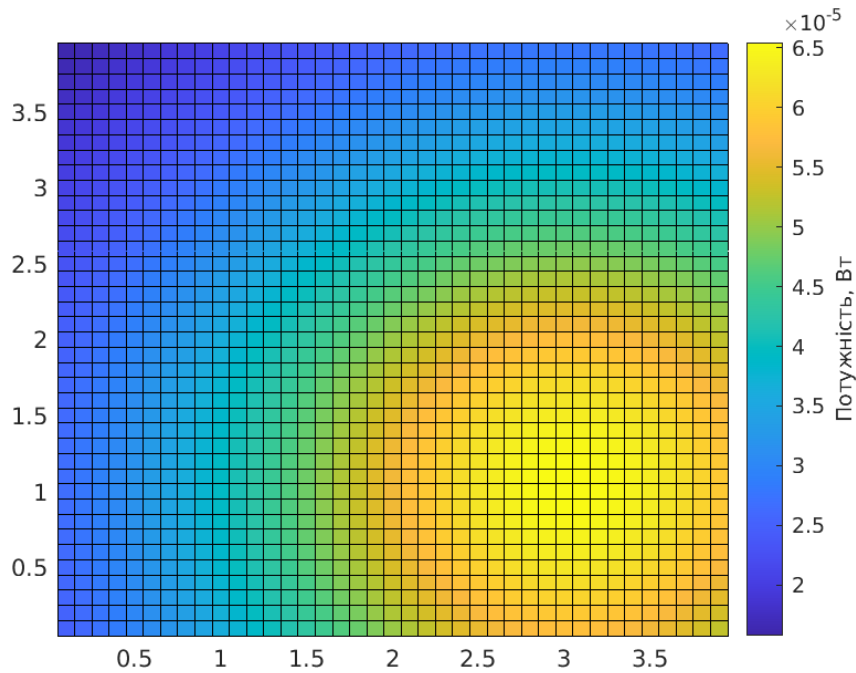


Рис. 3.4 Кольорова мапа потужності прийняття сигналу без спотворень від лампи L_3

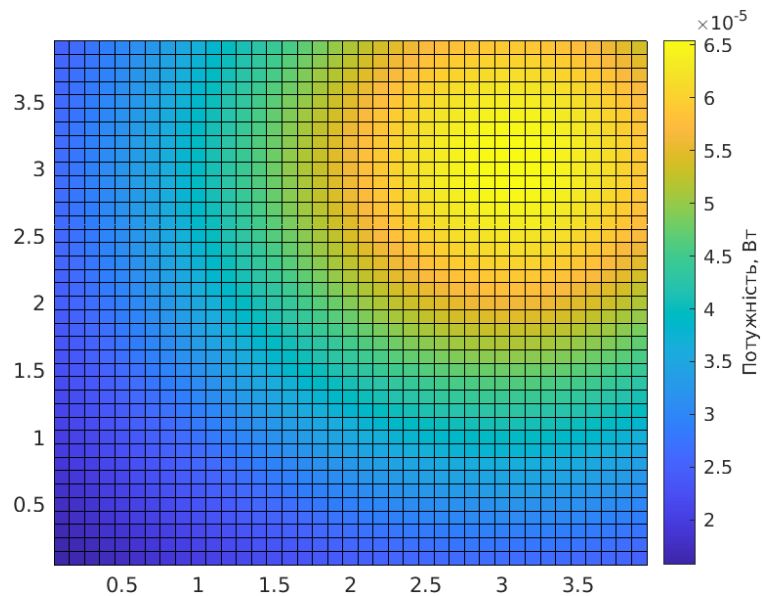


Рис. 3.5 Кольорова мапа потужності прийняття сигналу без спотворень від лампи L_4

Джерело: Розроблено автором

Використовуючи ці матриці можна визначити місцеположення об'єктів, порівнюючи значення потужності, прийняте об'єктом із значенням потужності у базі даних. При наявності у кімнаті лише 1 лампи, як зображено на рис. 3.1, визначити точне місцеположення об'єкта не можна, можна лише визначити

відстань від об'єкта до лампи. На рис. 3.1 можна побачити, що деякі клітинки мають подібні значення потужності. При використанні кількох ламп, наприклад, 4-х, як зображено на Рис. 2.3, можна визначити точне місцеположення об'єкта у приміщенні із застосуванням додаткових обчислень, із використанням більшої кількості даних, зображеної на Рис. 3.2-3.5.

3.2 Визначення ламп, сигнал від яких не спотворений

Як зазначено в підрозділі 1:3, на однаковій відстані від лампи на підлозі однакова потужність прийнятого сигналу. Отже для певного значення потужності який отримав шуканий об'єкт відповідає безліч точок на підлозі отже і в базі даних. Хто з однією лампою можливо визначити місцеположення об'єкта. Якщо взяти дві лампи то в нас буде два кола. у геометрії кола можуть перетинатися 0 1 або 2 рази. якщо прийнято сигналом без спотворень, то має бути принаймні одна точка, в якій сигнали від всіх ламп перетнуться, тобто прийнята потужність об'єкта відповідатиме потужності в базі даних для всіх ламп в певній координаті. якщо вам пишуть дві то може то в нас лише є два кола однакової потужності які перетинаються в одній або двох точках. в одній найчистіше відбуваються в тих випадках коли інша точка однакової відстані між обох ламп знаходилася б за межами кімнати і тому не потрапляє в нашу базу даних. в таких випадках набагато легше визначити місцеположення об'єкта якщо лише сигнал від цих двох ламп прийнято без спотворень, оскільки ми однозначно можемо сказати в за двома лампами яке значення найкраще підходить.

Як вказано в підрозділі 1:3, коли немає завад, то значення функції потужності сили від кожної лампи можуть відповідати певним координатам, які розташовані по колу. Ці кола однакової потужності, які відповідають потужності прийнятого сигналу користувачем, перетинаються в одній точці. Тобто є значення в базі даних Fingerprint, якому відповідає значення прийнятої потужності шуканим об'єктом, як зображено на Рис. 3.6

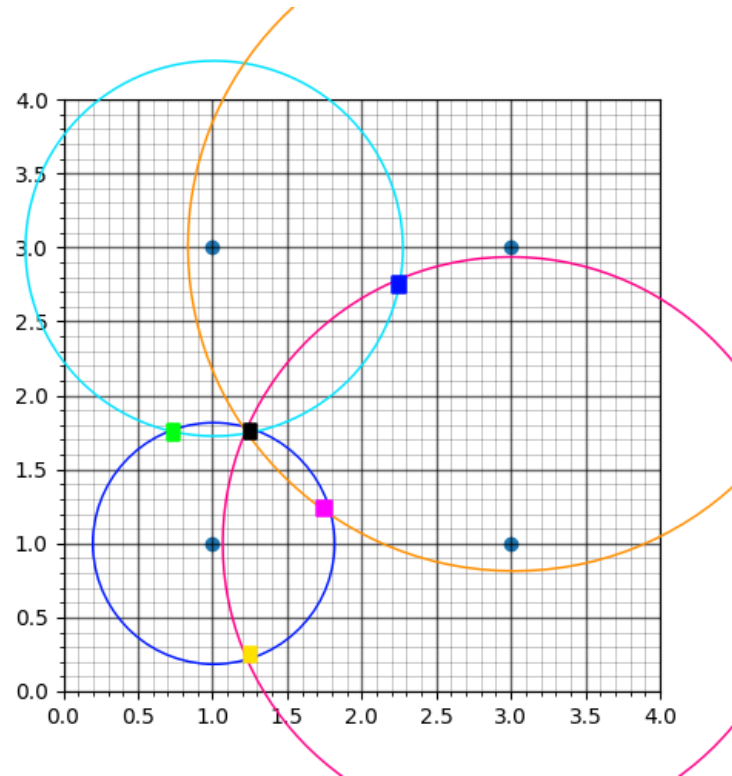


Рис. 3.6 Сигнал у чорній точці (1,25; 1,75) перетинається від усіх ламп та точно відповідає значенню користувача

Якщо є завади лише від однієї лампи, то в нас виникає ситуація, коли значення від трьох інших ламп перетинаються в одній точці, як зображено на 3.7.

Якщо сигнал від однієї лампи прийнято із спотворенням, а від трьох інших місцеположення об'єкта за цими термінами як зображено на Рис. 3.7

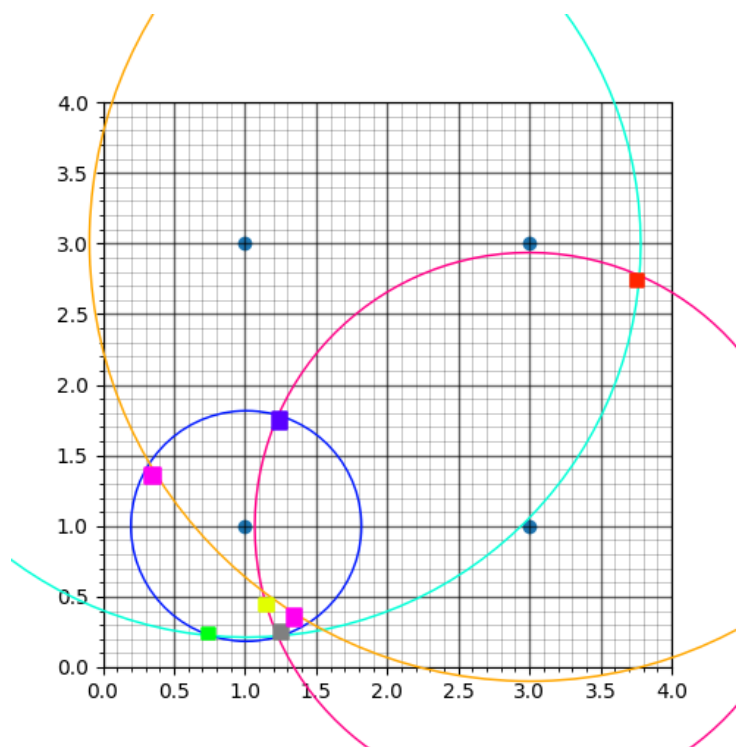


Рис. 3.7 Сигнал у сірій точці $(1,25; 0,25)$ перетинається від трьох ламп.

Значення цієї точки однозначно найкраще відповідають значенню користувача

Якщо прийнятий сигнал спотворено від двох ламп. При 4-х лампах в нас є шість пар ламп, кожна пара може мати від нуля до двох перетинів. Іноді завада може бути настільки сильною, що вона збільшує або зменшує прийнятий сигнал настільки, що таких значень немає в базі даних опорних точок Fingerprint. Насправді, будь-яка додатна завада перпендикулярно під лампою та будь-яка від'ємна завада в протилежному куті кімнати є такою. В таких випадках зрозуміло, що ця лампа спотворена і для пошуку місцеположення об'єкту варто використовувати сигнал від інших ламп. У роботі використано алгоритм визначення ламп, сигнал від яких прийнято без спотворень, який описано в [73]. Після знаходження визначення цих точок визначаємо точки, в яких сигнал хоча б від 2 ламп є схожим до потужностей сигналу користувача. Комбінація ламп, різниця потужностей в опорних точках яких найбільш наближена до користувача вважаємо за правильну. У випадку на Рис 3.8 однозначно зрозуміло, що сигнал від ламп, які знаходяться в лівій площині прийнято без спотворень, і тому перетин кіл однакової потужності цих ламп,

тобто значення прийнятої потужності шуканого об'єкта із значенням в базі даних опорних точок Fingerprint за цими двома лампами співпадають лише в одній зеленій точці (2,75; 1,75).

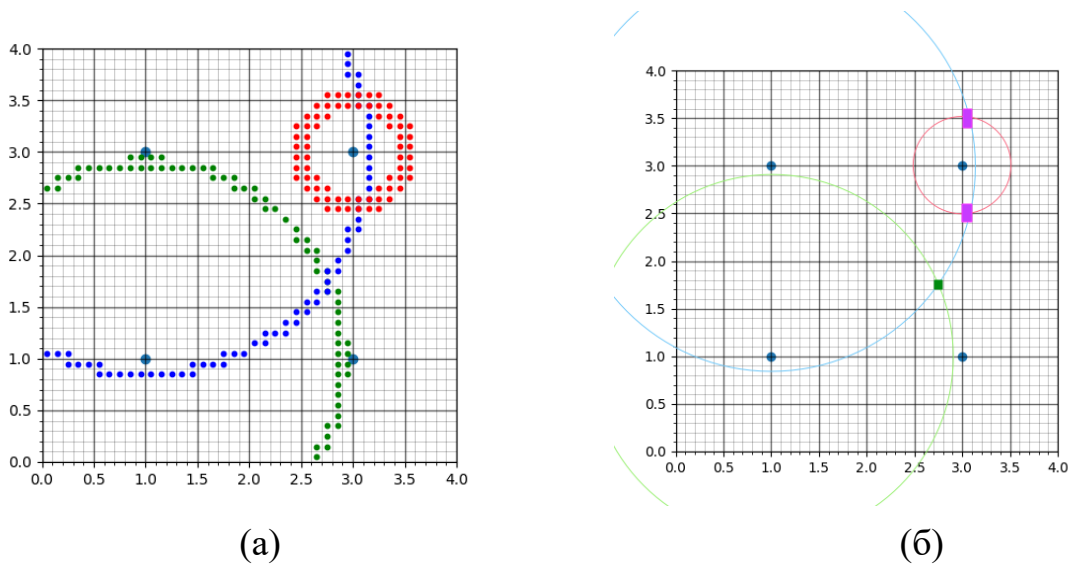


Рис. 3.8 Знаходження точок, потужність яких схожа із потужністю користувача (а) за допомогою програмних засобів, (б) вручну

Ситуація визначення місцеположення об'єкта на Рис. 3.8 стає тривіальною після визначення, що лампи у лівій півплощині неспотворені. На Рис. 3.9 цієї інформації недостатньо, оскільки є дві точки, в яких сигнал за двома лампами в базі даних опорних точок Fingerprint однаковий. Якщо припустити, що до точки, до якої ближче розташовані інші перетини кіл однакової потужності, правильна, то в такому випадку точка із координатами (1,25; 1,75) правильна.

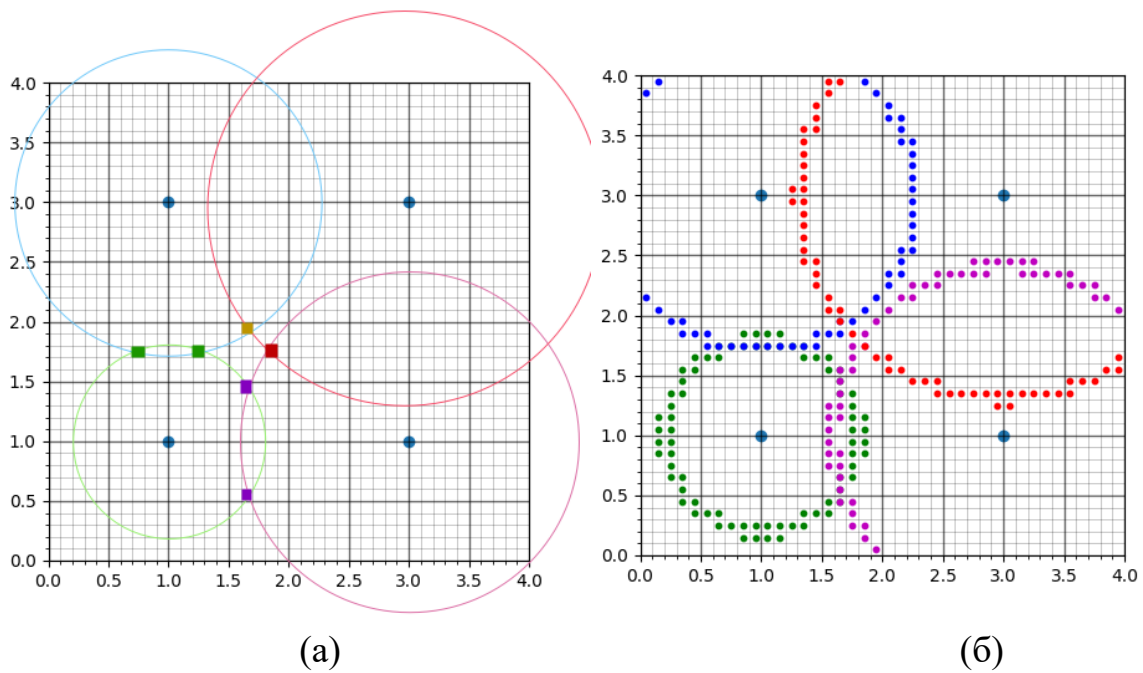


Рис. 3.9 Знаходження точок, потужність яких схожа із потужністю користувача (а) вручну, (б) за допомогою програмних засобів

В деяких випадках, це припущення не працює. На Рис. 3.10 сигнал координата шуканого об'єкта у $(0,75; 0,75)$, але таке припущення визначає координата об'єкта неправильно за $(1,25; 0,75)$, оскільки 2 точки ближче до неправильної координати, а 1 взагалі співпадає із неправильною точкою. До дійсного місцеположення об'єкта ближче лише одна точка. Подібна ситуація на Рис. 3.7, коли сигнал прийнято із спотвореннями від ламп, що знаходяться в лівій півплощини на 20% більше, за очікувану ($k=1.2$).

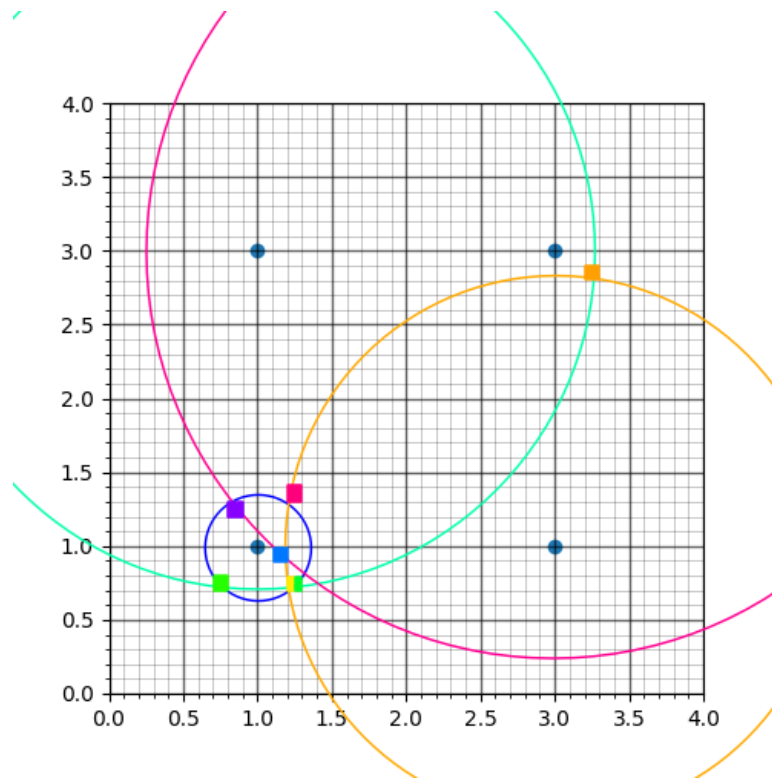


Рис. 3.10 Ситуація, коли сигнал від ламп у правій півплощини прийнято на 25% більше за еталонне значення ($k=1.2$) та шуканий об'єкт знаходиться у координатах $(0,75; 0,75)$

На Рис. 3.11 сигнал координата шуканого об'єкта у $(0,75; 1,25)$, але таке припущення визначає координата об'єкта неправильно за $(1,25; 1,25)$, оскільки 1 точки ближче до неправильної координати, і точка, в якій користувач не знаходиться, має еквівалентне значення до прийнято шуканим об'єктом по всім лампам. Жодна додаткова точка ближче дійсного місцеположення об'єкта. Подібна ситуація на Рис. 3.6, коли сигнал прийнято із спотвореннями від ламп, що знаходяться в лівій півплощини, на 20% більше за еталонне значення ($k=1.2$).

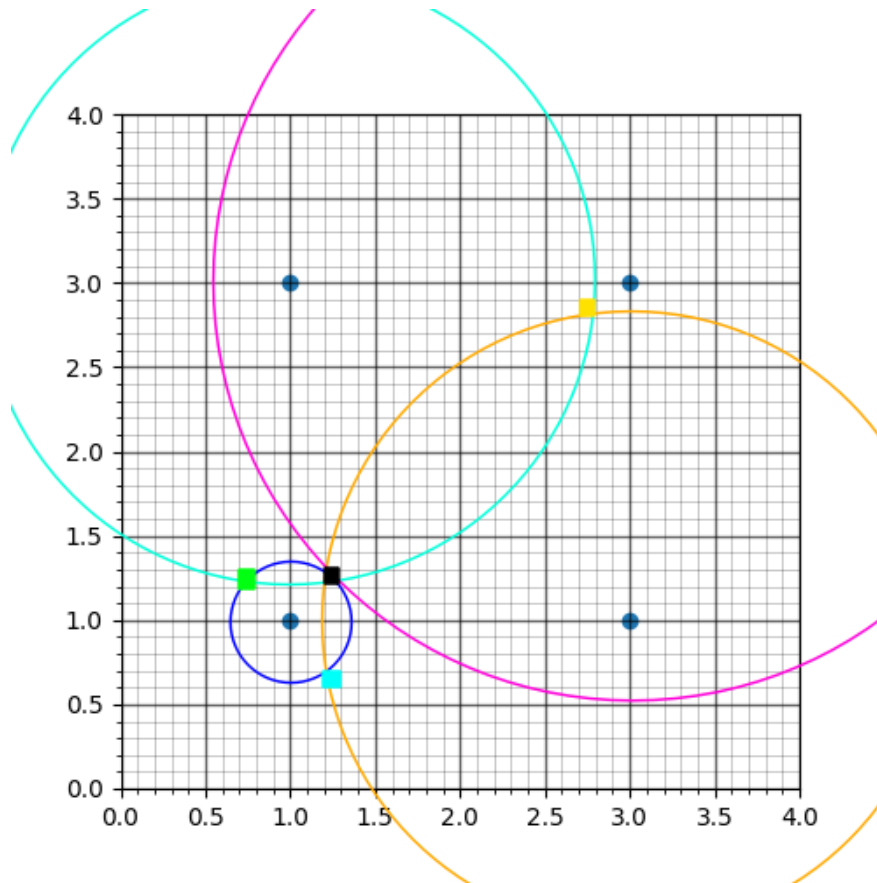


Рис. 3.11 Ситуація, коли сигнал від ламп у правій півплощини прийнято на 20% більше за еталонне значення ($k=1.2$) та шуканий об'єкт знаходиться у координатах (0,75; 1,25)

До цього розглядали лише певні точки. Зараз розглянемо одразу усі точки на підлозі. На Рис. 3.12 сигнал від ламп у правій півплощини кімнати спотворено із коефіцієнтом спотворення $k=1.2$. Тобто, прийнятий сигнал більший за еталонний на 20%. Червоним кольором позначено, де у результаті симуляції не вдалося (правильно) визначити місцеположення об'єкта.

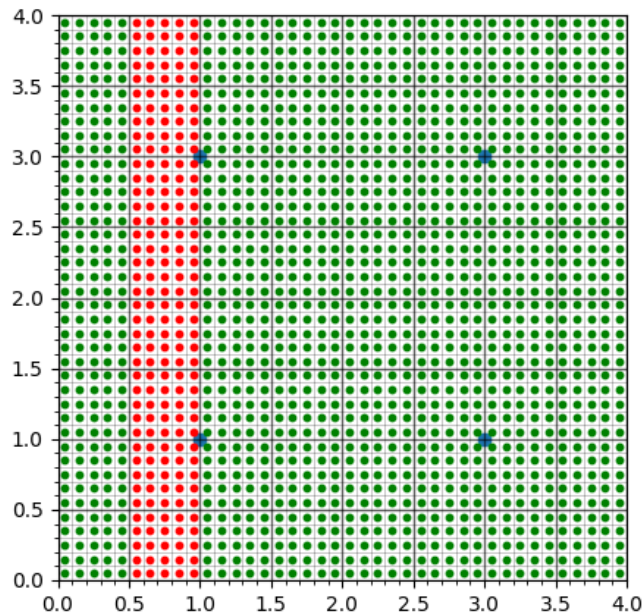


Рис. 3.12 Правильність визначення координат, коли сигнал від ламп L_1L_2 не спотворений, а $L_3 L_4$ спотворений. $k=1,2$, потужність прийнятого сигналу на 20% більша за еталонну

Як можна побачити, при координатах $x=0,5\dots 1$, місцеположення об'єкта визначається неправильно. На Рис. 3.6, 3.7, 3.9-3.11 показано, що якщо об'єкт знаходиться ближче до лівої стіни, ніж лампи ($x < 1$), та прийнято сигнал від ламп, що знаходяться справа із додатньою завадою, то ситуація стає аналогічній, якби сигнал було б прийнято без завад, але користувач знаходився симетрично в іншій точці.

Використовуючи зазначений метод потенційних точок за двома лампами, але коли прийнято сигнал із спотворенням не від прилеглих, а від ламп, що лежать по діагоналі, побудовано графік 3.13.

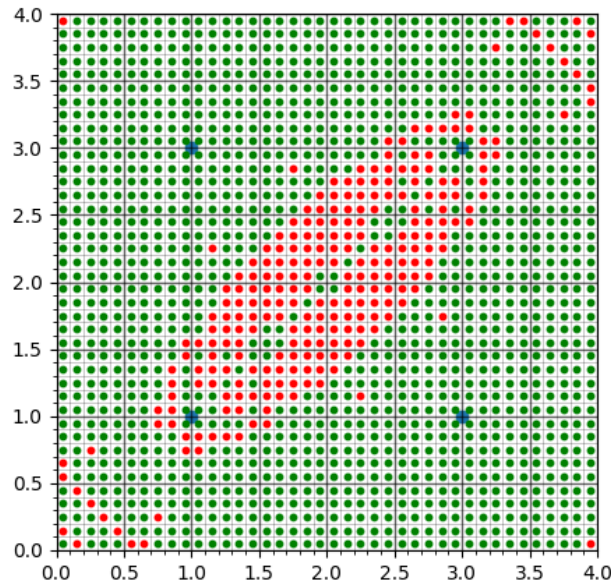


Рис. 3.13 Правильність визначення координат, коли сигнал від ламп L_1L_4 не спотворений, а $L_2 L_3$ спотворений

Можна побачити, що є значне спотворення при використанні допоміжних точок серед пар ламп. Використовуючи допоміжні точки за 3а та 4-а лампами, як описано у підрозділі 2.3, тобто використовуючи алгоритм із [Дипломна робота], можна отримати набагато кращі результати, у яких точність при випадках, коли лампи розміщено діагонально, сигнал від яких прийнято із спотворенням визначає правильно $\geq 98\%$ випадках.

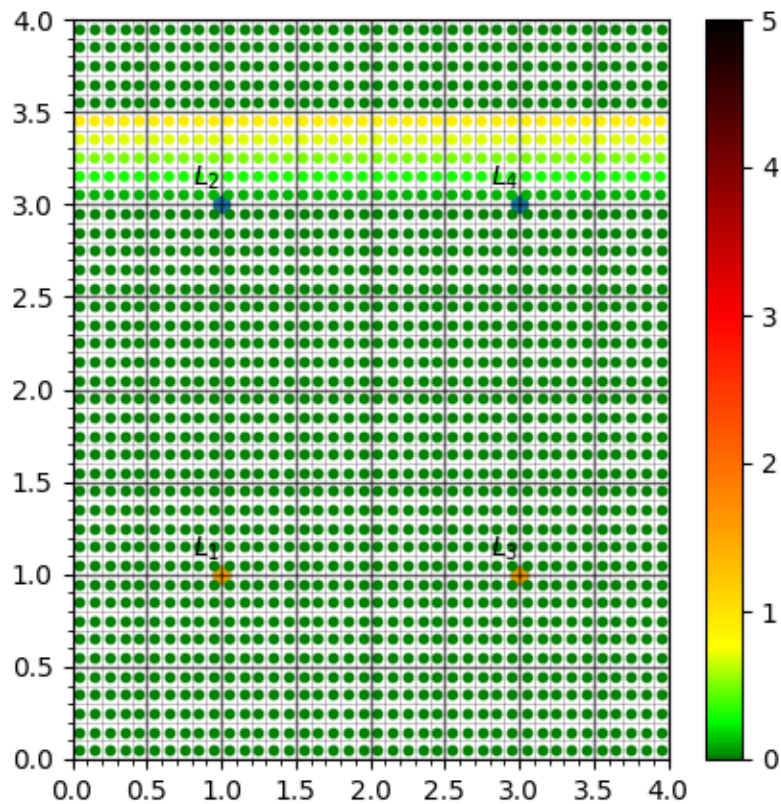


Рис. 3.14. Правильність визначення координат шуканого об'єкта. Сигнал від ламп L_1 та L_3 спотворено, а від L_2 та L_4 – ні. Відхилення в метрах відповідає кольору зліва від самої діаграми

На Рис. 3.14 можна побачити, що якщо інша прилегла пара ламп спотворена, то загалом ситуація симетрична. На Рис. 3.15 зображено випадок, коли прийнята потужність менша за еталону на 10%. Кількість помилок менша, та помилки знаходяться в протилежну сторону, до центру кімнати від ламп, сигнал від яких прийнято без спотворень

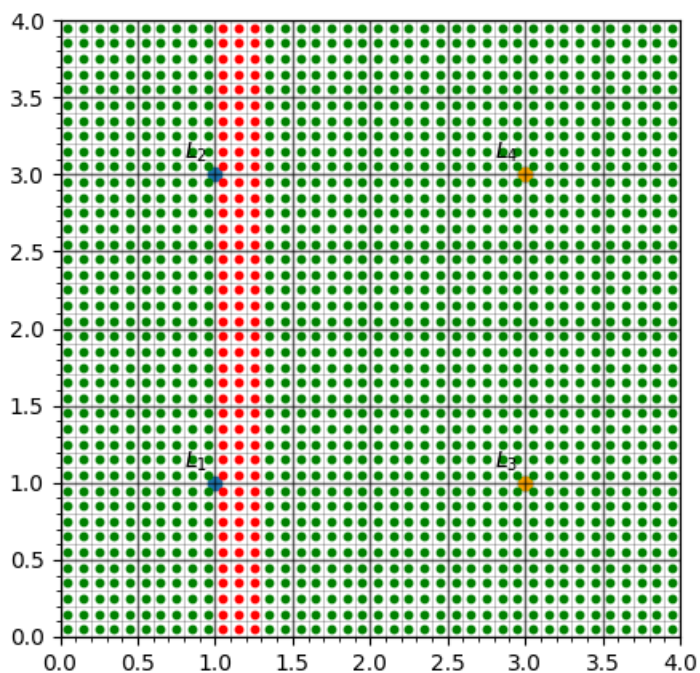


Рис. 3.15 Правильність визначення координат, коли сигнал від ламп L_1L_2 не спотворений, а $L_3 L_4$ спотворений. $k=0,9$, потужність прийнятого сигналу на 10% менша за еталонну

3.3 Визначення координат об'єкта, із використанням інформації про спотвореність сигналу від ламп

Але бувають випадки, коли від однієї комбінації ламп в двох різних точках однакова потужність, як на Рис. 3.1. Для таких випадках розроблено наступну частину програмних засобів.

До цього розглядалися випадки, коли коефіцієнт спотворення має певне значення та при певній конфігурації кімнати: Лампи розташовані квадратом, на один метр від стіни та два метри між прилеглими лампами. На Рис. 3.16 зображено ситуацію, коли прилеглі лампи спотворено, а лампи розташовані ближче до центру кімнати, 1,5 метри від стін та 1 метр між лампами, а на Рис 3.17, коли лампи по діагоналі спотворено. На Рис. 3.18 побудовано графік залежності вірогідності правильного визначення місцеположення об'єктів від

коефіцієнта спотворення k для конфігурацій, зображених на Рис. 3.16 та Рис. 3.17

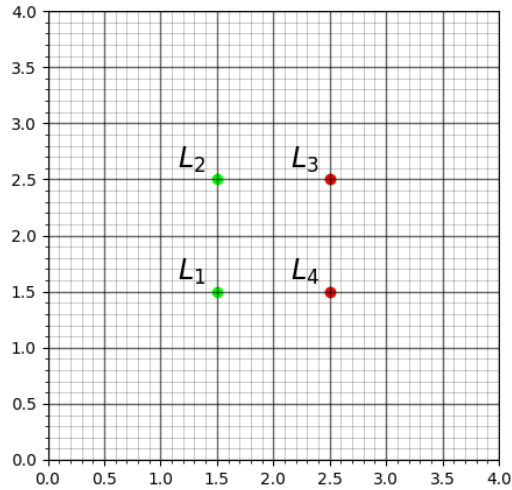


Рис. 3.16 Прилеглі лампи спотворено. Відстань між лампами 1 метр

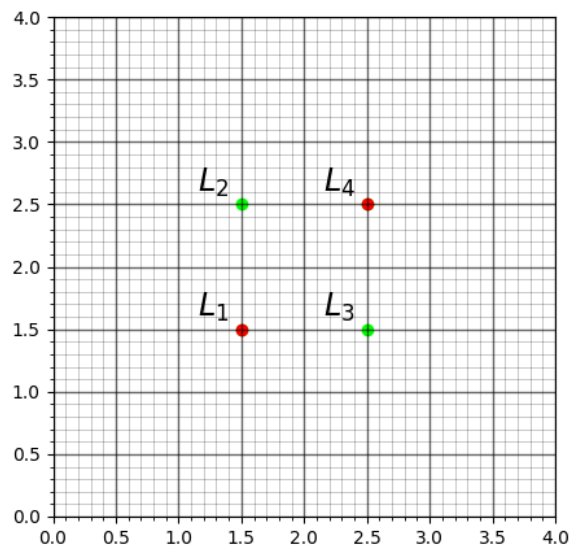


Рис. 3.17 Діагональні лампи спотворено. Відстань між лампами 1 метр

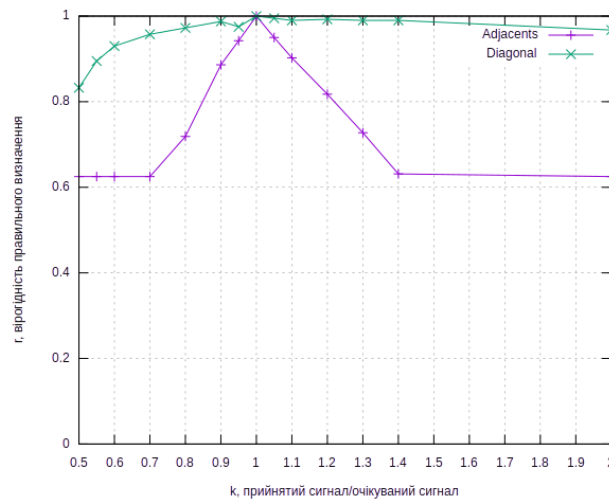


Рис. 3.18 Залежність вірогідності правильного визначення місцеположення об'єктів при відстані між лампами 1 метр. Фіолетовим кольором позначено прилеглі лампи (Рис. 3.16), Зеленим кольором позначено діагональні лампи (Рис. 3.17)

На Рис. 3.19 зображено ситуацію, коли прилеглі лампи спотворено, а лампи розташовані на помірній відстані, 1 метри від стін та 2 метр між лампами, а на Рис 3.20, коли лампи по діагоналі спотворено. На Рис. f6 побудовано графік залежності вірогідності правильного визначення місцеположення об'єктів від коефіцієнта спотворення k для конфігурацій, зображених на Рис. 3.19 та Рис. 3.20

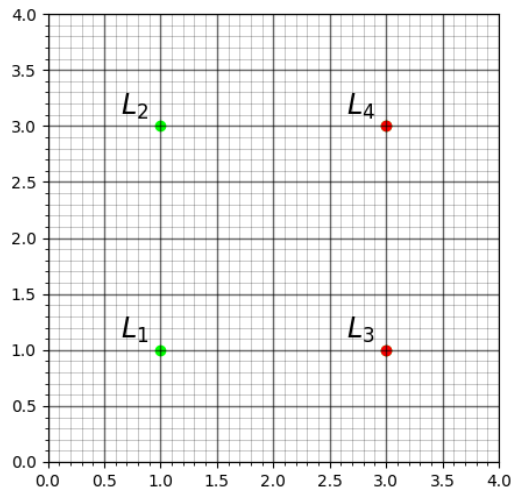


Рис. 3.19 Прилеглі лампи спотворено. Відстань між лампами 2 метр

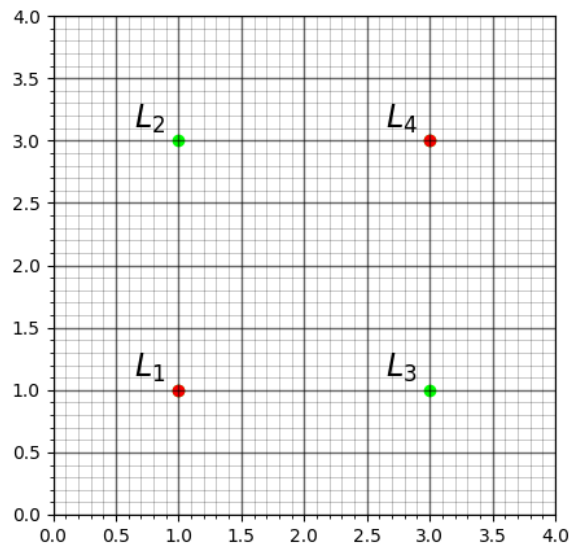


Рис. 3.20 Діагональні лампи спотворено. Відстань між лампами 2 метр

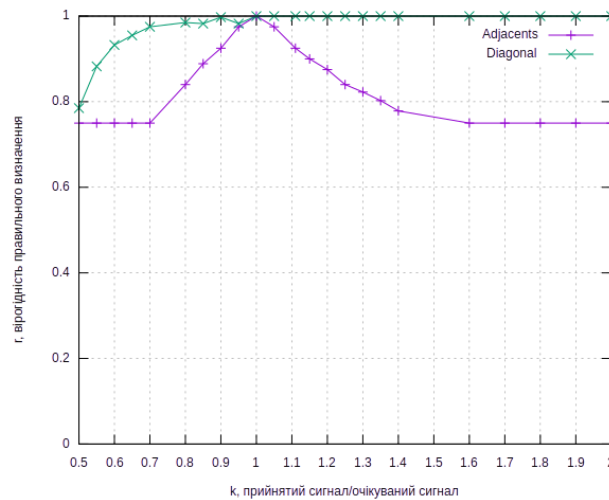


Рис. 3.21 Залежність вірогідності правильного визначення місцеположення об'єктів при відстані між лампами 2 метр. Фіолетовим кольором позначено прилеглі лампи (Рис. 3.19), Зеленим кольором позначено діагональні лампи (Рис. 3.20)

На Рис. 3.22 зображено ситуацію, коли прилеглі лампи спотворено, а лампи розташовані ближче до країв кімнати, 0,5 метрів від стін та 2,5 метр між лампами, а на Рис 3.23, коли лампи по діагоналі спотворено. На Рис. 3.24 побудовано графік залежності вірогідності правильного визначення місцеположення об'єктів від коефіцієнта спотворення k для конфігурацій, зображених на Рис. 3.22 та Рис. 3.23

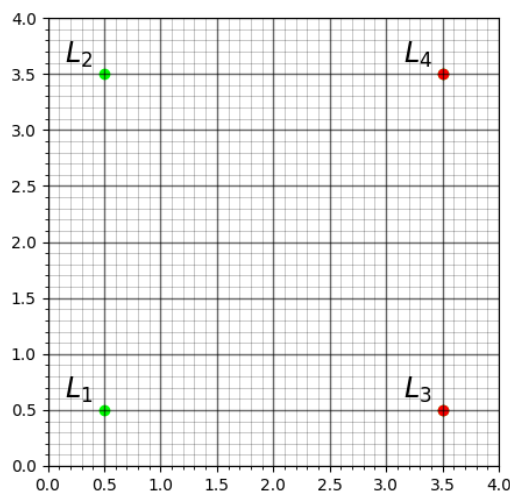


Рис. 3.22 Прилеглі лампи спотворено. Відстань між лампами 3 метр

Джерело: Розроблено автором

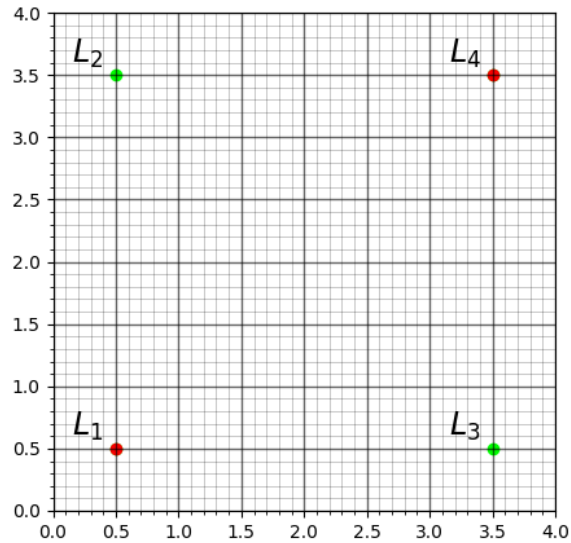


Рис. 3.23 Діагональні лампи спотворено. Відстань між лампами 3 метр

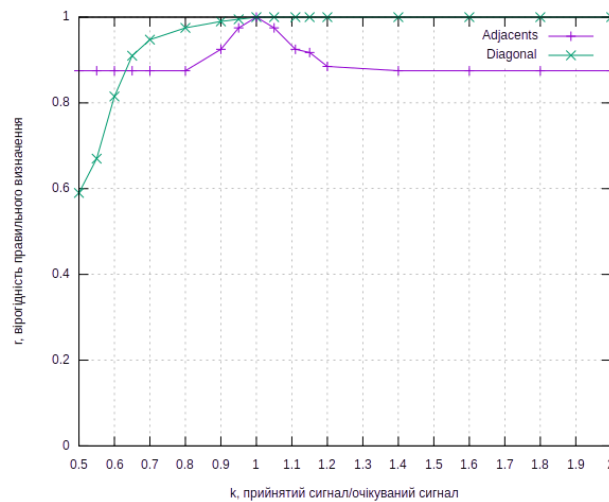


Рис. 3.24 Залежність вірогідності правильного визначення місцеположення об'єктів при відстані між лампами 3 метр. Фіолетовим кольором позначено прилеглі лампи (Рис. 3.22), Зеленим кольором позначено діагональні лампи (Рис. 3.23)

Отже, як можна побачити, що для всіх конфігурацій коли лампи, сигнал від яких прийнято із спотворенням, розташовані прилегле одна до одної, то по мірі віддалення коефіцієнту спотворення прийнятого сигналу k від 1, причому як в більшу так і в меншу сторону, правильність визначення місцеположення спадає до певного рівня, а потім залишається сталим. При спотворенні від ламп,

що розташовані по діагоналі одна від одної, при коефіцієнті $k > 1$ вірогідність правильного визначення місцеположення наближається до 1. При $k < 1$, вона спадає з пришвидшенням. Із Рис 3.18, 3.21 та 3.24 видно, що чим ближче лампи розташовані до країв кімнати, тим краща ймовірність визначити місцеположення при спотворенні сигналу від прилеглих ламп, але гірша ймовірність визначити місцеположення, якщо сигнал від ламп по діагоналі спотворено.

Висновки

Для вирішення завдання локалізації було сформовано базу даних розподілу потужності сигналу, прийнятого від чотирьох джерел LED-випромінювання. Отримані дані представлені у вигляді тривимірного масиву, де значення потужності корелюють із просторовими координатами (x, y) та індексом відповідної лампи L.

Проведено пошук об'єкта в приміщенні шляхом мінімізації середньоквадратичної помилки між еталонними значеннями з бази даних опорних точок Fingerprint та фактично отриманими показниками потужності шуканим об'єктом.

Якщо певна точка демонструє мінімальне відхилення для 4 або 3 ламп, вона з високою ймовірністю визначається як істинне місцеположення об'єкта.

У разі відсутності єдиної точки збігу проводиться аналіз пар ламп, сигнали від яких не зазнали суттєвих спотворень. Якщо результатом такого аналізу є лише одна точка, то її і вважаємо за місцерозташування об'єкта

Якщо за результатами вимірювань виникає дві потенційні локації, застосовується метод «допоміжних точок». Положення об'єкта визначається за принципом найменшої відстані до допоміжних точок, отриманих від джерел із найменшим рівнем інтерференції.

Однією з основних проблем є імовірність виникнення помилкових координат через спотворення сигналу, що призводить до ідентифікації значень, які існують у базі даних опорних точок Fingerprint для інших точок простору.

Для мінімізації таких випадків проведено дослідження щодо зміни конфігурації розташування ламп на стелі.

Виявлено, що розміщення передавачів ближче до стін підвищує ймовірність визначення правильного місцеположення об'єкта за умови спотворення сигналів від прилеглих ламп.

Ймовірність визначення місцеположення об'єкта залишається високою при спотворенні сигнал від ламп, що розташовані діагонально одна до одної, але при розміщенні ламп ближче до стін незначно погіршує цей показник.

ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ

Технологія Li-Fi – дуже перспективна технологія, діапазон електромагнітного спектру у 10'000 разів більше за увесь радіодіапазон, тому і пропускна здатність набагато більша. Прогнозується широке використання цієї технології. Вона також набагато точніша у позиціонуванні. Одна із невирішених проблем залишається завади, які особливі саме для видимого світла. Для вирішення поставленої задачі обрано методи RSS та Fingerprint.

Розроблено алгоритми пошуку об'єктів у приміщенні. Якщо виявлено, що лише сигнал від двох ламп прийнято без спотворень, то спочатку визначаємо, які із ламп не спотворено. Після визначення цих ламп, знаходимо координати, середньоквадратичне відхилення із потужністю, прийнятою шуканим об'єктом найменше. Якщо така точка одна, то вважаємо її за місцеположення об'єкта. Якщо їх дві, то вирішуємо задачу визначення, у якій із цих двох «потенційних» точок знаходиться об'єкт використовуючи метод «допоміжних точок». До якої із цих «потенційних» точок більше «допоміжних» точок, ту і вважаємо за правильну.

Дослідження показало, що використання у якості допоміжних точок як результат позиціонування пар ламп, у яких одна пара спотворена, а інша ні не є ефективною методикою. При використанні результатів позиціонування за 3-а та 4-а точками показує більшу вірогідність правильно визначити місцеположення об'єкта.

В деяких випадках завада впливає на потужність сигналу так, що прийнята потужність шуканого об'єкту відповідає значенню в базі даних, яка лежить в іншій координаті. Для вирішення цієї проблеми було змінено конфігурацію ламп: Ближче до та далі від центру.

Виявлено, що розміщення передавачів ближче до стін підвищує ймовірність визначення правильного місцеположення об'єкта за умови спотворення сигналів від прилеглих ламп.

Ймовірність визначення місцеположення об'єкта залишається високою при спотворенні сигнал від ламп, що розташовані діагонально одна до одної, але при розміщені ламп ближче до стін незначно погіршує цей показник.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Романов О.І, Мікляєв Г.О. Внутрішнє позиціонування за пошуком найближчого значення RSS використовуючи технологію Li-Fi. «Перспективи телекомунікацій» 16-19 квітня 2024 р., м. Київ, Україна.
2. Wang K, Liu Y, Hong Z. RSS-based visible light positioning based on channel state information. *Opt Express*. 2022 Feb 14;30(4):5683-5699. doi: 10.1364/OE.451209. PMID: 35209525.
3. Romanov, O., Nesterenko, M., Boggia, G., Striccoli, D. Construction and Methods for Solving Problems at the SDN Control Level // *Lecture Notes in Electrical Engineering*, 2023, 965 LNEE, p. 85–101, URL: https://link.springer.com/chapter/10.1007/978-3-031-24963-1_6
4. Bilt Industries INC, Warehouse Robots 101: We're Not Talking About the Future, [Heather Demand](#), URL: <https://www.biltindustries.com/warehouse-robots-101/>
5. Wang, Kaiyao, Yi He, Xinpeng Huang, and Zhiyong Hong. 2023. "Adaptive Weighted K-Nearest Neighbor Trilateration Algorithm for Visible Light Positioning" *Photonics* 10, no. 3: 319. URL: <https://doi.org/10.3390/photonics10030319>
6. Salman, K., and E. Hamza. "Indoor positioning systems based on Li-Fi technology using RSS-triangulation with assisted by DNN." *International Journal of Intelligent Engineering & Systems* 15.2 (2022).
7. Romanov, O., Nesterenko, M., Marinov, A., Skolets, S., Burlaka, H. SDN network modeling using the GUI MiniEdit/Proceedings – 16th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering, TCSET 2022, 2022, стр. 637–642, DOI: 10.1109/TCSET55632.2022.9766897
8. Romanov, O., Dong, T.T., Nesterenko, M. The possibilities for deployment eco-friendly indoor wireless networks based on LiFi technology/Proceedings of

International Conference on Applied Innovation in IT, 2020, 8(1), стр. 41–48, URL: [https://www.scopus.com/inward/record.uri?eid=2-s2.0-](https://www.scopus.com/inward/record.uri?eid=2-s2.0-85085966325&partnerID=40&md5=abf8759d141835b8fcd8082fdb4b04dc)

85085966325&partnerID=40&md5=abf8759d141835b8fcd8082fdb4b04dc

9. Romanov, O.I., Hordashnyk, Y.S., Dong, T.T. Method for calculating the energy loss of a light signal in a telecommunication Li-Fi system/2nd International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2017 – Proceedings, 2017, 8095404, DOI: 10.1109/UkrMiCo.2017.8095404

10. Oleksandr Romanov, Hrigorii Burlaka, Mankivskyi Vladimir, Mikliaiev Hlieb, Directions for the development of SDN networks: setting tasks, management methods, interfaces, IEEE UkrMiCo-2023

11. Багатовузлова мережа SDN з використанням технології Li-Fi. Піталова, М. Д., Мікляєв, Г. О. Матеріали чотирнадцятої Міжнародної науково-технічної конференції «Перспективи телекомунікацій» 13-17 квітня 2020 р., м. Київ, Україна. URL: <http://conferenc.its.kpi.ua/proc/article/view/200883>

12. Засади проектування офісної мережі на базі технології LI-FI. Мікляєв Г. О. Матеріали п'ятнадцятої Міжнародної науково-технічної конференції «Перспективи телекомунікацій» 12-16 квітня 2021 р., м. Київ, Україна, 129–131. URL: <http://conferenc.its.kpi.ua/proc/article/view/230401>

13. Г. О. Мікляєв, О. І. Романов Throughput increase in the office LI-FI system. Information and Telecommunication Sciences, № 2 (2021), URL: <https://doi.org/10.20535/2411-2976.22021.16-21>

14. Petrosino A., Striccoli, D., Romanov, O., Boggia, G., Grieco, L.A. Fidelity for Internet of Things: A survey/ Optical Switching and Networking, Volume 48, March 2023, 100732 URL: <https://doi.org/10.1016/j.osn.2023.100732>

15. Romanov, O., Siemens, E., Nesterenko, M., Mankivskyi, V. Mathematical description of control problems in SDN networks/Proceedings of International

Conference on Applied Innovation in IT, 2021, 9(1), стр. 33–39, URL: [https://www.scopus.com/inward/record.uri?eid=2-s2.0-](https://www.scopus.com/inward/record.uri?eid=2-s2.0-85109068329&partnerID=40&md5=adbcc806191b30c911f8fa4c52eec604)

[85109068329&partnerID=40&md5=adbcc806191b30c911f8fa4c52eec604](https://www.scopus.com/inward/record.uri?eid=2-s2.0-85109068329&partnerID=40&md5=adbcc806191b30c911f8fa4c52eec604)

16. Romanov, O., Nesterenko, M., Veres, L., Kamarali, R., Saychenko, I. Methods for calculating the performance indicators of ip multimedia subsystem (ims)/(2021) Lecture Notes in Networks and Systems, 152, pp. 229-256. DOI: 10.1007/978-3-030-58359-0_13

17. Petrosino A., Striccoli, D., Romanov, O., Boggia, G., Grieco, L.A. Fidelity for Internet of Things: A survey/ Optical Switching and Networking, Volume 48, March 2023, 100732 URL: <https://doi.org/10.1016/j.osn.2023.100732>

18. V. Jungnickel et al., "LiFi for Industrial Wireless Applications," 2020 Optical Fiber Communications Conference and Exhibition (OFC), San Diego, CA, USA, 2020, pp. 1-3.

19. C. Xie, W. Guan, Y. Wu, L. Fang and Y. Cai, "The LED-ID Detection and Recognition Method Based on Visible Light Positioning Using Proximity Method," in IEEE Photonics Journal, vol. 10, no. 2, pp. 1-16, April 2018, Art no. 7902116, doi: 10.1109/JPHOT.2018.2809731.

20. E. W. Lam and T. D. C. Little, "Indoor 3D Localization with Low-Cost LiFi Components," 2019 Global LIFI Congress (GLC), Paris, France, 2019, pp. 1-6, doi: 10.1109/GLC.2019.8864119.

21. K. D. Salman and E. K. Hamza, "Performance Li-Fi Indoor Positioning Systems using RSS-Triangulation Method," 2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS), Coimbatore, India, 2022, pp. 1741-1746, doi: 10.1109/ICAIS53314.2022.9742953.

22. M. Müller, D. Behnke, P. -B. Bök, C. Kottke, K. L. Bober and V. Jungnickel, "Leverage LiFi in Smart Manufacturing," 2020 IEEE Globecom

Workshops (GC Wkshps, Taipei, Taiwan, 2020, pp. 1-6, doi: 10.1109/GCWkshps50303.2020.9367446.

23. X. Qi, L. Du, S. Wang and J. Liu, "Design of the Indoor Positioning Glasses for the Blind Based on the LiFi Technology," 2018 IEEE 4th International Conference on Computer and Communications (ICCC), Chengdu, China, 2018, pp. 744-747, doi: 10.1109/CompComm.2018.8780945.

24. H. Vijayaraghavan, J. Von Mankowski, C. Mas-Machuca and W. Kellerer, "PlaciFi: Orchestrating Optimal 3D Access Point Placement for LiFi-WiFi Heterogeneous Networks," in *IEEE Access*, vol. 11, pp. 115415-115429, 2023, doi: 10.1109/ACCESS.2023.3325097.

25. K. D. Salman and E. K. Hamza, "Performance Li-Fi Indoor Positioning Systems using RSS-Triangulation Method," 2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS), Coimbatore, India, 2022, pp. 1741-1746, doi: 10.1109/ICAIS53314.2022.9742953.

26. E. W. Lam and T. D. C. Little, "Indoor 3D Localization with Low-Cost LiFi Components," 2019 Global LIFI Congress (GLC), Paris, France, 2019, pp. 1-6, doi: 10.1109/GLC.2019.8864119.

27. C. Xie, W. Guan, Y. Wu, L. Fang and Y. Cai, "The LED-ID Detection and Recognition Method Based on Visible Light Positioning Using Proximity Method," in *IEEE Photonics Journal*, vol. 10, no. 2, pp. 1-16, April 2018, Art no. 7902116, doi: 10.1109/JPHOT.2018.2809731.

28. Kawther Dawood Salman, Ekhlas Kadhum Hamza, "Indoor Positioning Systems Based on Li-Fi Technology Using RSS-Triangulation With Assisted By DNN", *International Journal of Intelligent Engeeniring & Systems*, 2021 DOI: 10.22266/ijies2021

29. Mohammad Thiab, Ahmad Al-Harbi, Mohammad A Ghalib "Indoor Positioning System for Blind using Li-Fi", *College of Engineering & Applied*

Sciences, Department of Engineering (ENGR), CPEG 480 – Senior Design Capstone II, Fall 2019.

30. Romanov, O.I., Fediushyna, D.M., Dong, T.T. Model and method of Li-Fi network calculation with multipath light signals/(2018) 2018 International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2018, DOI: 10.1109/UkrMiCo43733.2018.9047550

31. Romanov, O.I., Hordashnyk, Y.S., Dong, T.T. Method for calculating the energy loss of a light signal in a telecommunication Li-Fi system (2017) 2nd International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2017 – Proceedings, стаття № 8095404

32. PureLiFi, LiFi system LiFi-XC, URL: <https://www.purelifi.com/products/lifi-xc/>

33. Romanov O., Korniienko N., Obod I., Svyd I. Construction of the SDN Control Level Based on ONOS./ 2021 IEEE International Conference on Information and Telecommunication Technologies and Radio Electronics (UkrMiCo), Odesa, Ukraine, 2021, pp. 127-132, doi: 10.1109/UkrMiCo52950.2021.9716691.

34. Romanov O., Burlaka H., Mankivskiy V., Mikliaiev H. Directions for the Development of SDN Networks: Setting Tasks, Management Methods, Interfaces/2023 IEEE International Conference on Information and Telecommunication Technologies and Radio Electronics (UkrMiCo).

DOI: 10.1109/UkrMiCo61577.2023.10380413

35. Романов, О., Мікляєв, Г.; . ПРОГНОЗУВАННЯ СПОТВОРЕНИХ СИГНАЛІВ ВІД LED ЛАМП В СИСТЕМАХ ПОЗИЦІОНУВАННЯ ВСЕРЕДИНІ ПРИМІЩЕНЬ. Міжнародна Науково-технічна Конференція "ПЕРСПЕКТИВИ ТЕЛЕКОМУНІКАЦІЙ", Київ, 14-18 квітня 2025 р.

36. Razaque, A., Bektemyssova, G., Khan, M. *et al.* Comprehensive analysis of Li-Fi technology: positioning algorithms, security vulnerabilities, and future IoT

applications. *Telecommun Syst* **88**, 130 (2025). <https://doi.org/10.1007/s11235-025-01358-z>

37. Z. Mustafa and E. K. Hamza, "Enhancing Indoor Positioning Accuracy using a Hybrid Li-Fi/Wi-Fi System with Deep Learning Support", *Eng. Technol. Appl. Sci. Res.*, vol. 15, no. 2, pp. 21575–21585, Apr. 2025.

38. B. Salinas et al., "Experimental Characterization of LiFi Performance in Indoor Scenarios: Impact of Spatial Positioning and Multi-User Load," 2025 IEEE Latin-American Conference on Communications (LATINCOM), Antigua, Guatemala, 2025, pp. 225-230, doi: 10.1109/LATINCOM67778.2025.11345396.

39. Siddhi, Shalini and Roy, Stuti and Pandit, Shweta, "BLi-Fi: Li-Fi-Based Home Navigation for Blind", 2025 International Journal of Sensors Wireless Communications and Control, 2025, vol. 15 no. 2 pp 173-183 <https://doi.org/10.2174/0122103279302588240821042940>

40. Panda, B., Beria, H. & Pradhan, C. Deployment of Li-Fi in indoor positioning systems. *Int. j. inf. tecnol.* **13**, 123–130 (2021). <https://doi.org/10.1007/s41870-020-00485-x>

41. N. J. Sattigiri, G. Patel, S. Mondal, R. Pal and S. Prince, "Li-Fi based Indoor Positioning System," 2019 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2019, pp. 0290-0293, doi: 10.1109/ICCSP.2019.8697974.

42. Z. Ma et al., "LiFi Positioning and Optimization in an Indoor Factory Environment," IECON 2022 – 48th Annual Conference of the IEEE Industrial Electronics Society, Brussels, Belgium, 2022, pp. 1-6, doi: 10.1109/IECON49645.2022.9968530.,

43. Pathak, P., Sharavanan, S., Karthikeyan, M. *et al.* Enhanced optical visible light positioning via RSS-based evolutionary optimization under LOS and NLOS conditions. *Sci Rep* **15**, 42025 (2025). <https://doi.org/10.1038/s41598-025-27965-z>

44. Gandal, J.R., Tiwari, B., Katre, G., Ambrale, A. (2026). LiFi-WiFi Hybrid Systems for Industrial Fast Data Transfer. In: Tuba, M., Akashe, S., Joshi, A. (eds) ICT Systems and Sustainability. ICT4SD 2025. Lecture Notes in Networks and Systems, vol 1645. Springer, Cham. https://doi.org/10.1007/978-3-032-06662-6_48
45. M. T. Salman, D. R. Siddle and A. G. Udu, "Machine Learning Approach to Predict the DC Bias for Adaptive OFDM Transmission in Indoor Li-Fi Applications," in *IEEE Access*, vol. 13, pp. 9627-9641, 2025, doi: 10.1109/ACCESS.2025.3527205.
46. Naser, S., Alhussein, O. & Muhaidat, S. Towards multi-modal and cross-modal integration in LiFi-based sensing. *Sci Rep* **16**, 6038 (2026). <https://doi.org/10.1038/s41598-026-36891-7>
47. Y. Huang et al., "Energy-Efficient RIS-Aided Laser-Based LiFi System with Dynamic Coverage Optimization," ICC 2025 - IEEE International Conference on Communications, Montreal, QC, Canada, 2025, pp. 1-7, doi: 10.1109/ICC52391.2025.11161959.
48. S. Bastiaens, M. Alijani, W. Joseph and D. Plets, "A Review of the Quality of Positioning Service of VLP Within the Indoor Positioning System Landscape," in *IEEE Transactions on Instrumentation and Measurement*, vol. 75, pp. 1-33, 2026, Art no. 7000733, doi: 10.1109/TIM.2026.3655932.
49. M. Jahnke, B. Palmer, L. Perschin, E. Stoll and U. Kulau, "Light Simulation Framework for the Optimization of Spatial Structures for Intra-Satellite LiFi-Communication," 2025 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE), Halifax, NS, Canada, 2025, pp. 1-6, doi: 10.1109/WiSEE57913.2025.11229870.
50. N R Krishnamoorthy *et al* 2021 *J. Phys.: Conf. Ser.* **1770** 012030 DOI 10.1088/1742-6596/1770/1/012030

51. X. Wu, M. D. Soltani, L. Zhou, M. Safari and H. Haas, "Hybrid LiFi and WiFi Networks: A Survey," in *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1398-1420, Secondquarter 2021, doi: 10.1109/COMST.2021.3058296.

52. M. A. Afzal, D. He, Z. Zhu and Y. Yang, "Performance Evaluation of Wi-Fi Bluetooth Low Energy & Li-Fi Technology in Indoor Positioning," 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP), Shanghai, China, 2018, pp. 1-5, doi: 10.1109/ICDSP.2018.8631602.

53. Kassahun, E., Salau, A.O., Jember, K.A. *et al.* Efficient resource allocation scheme for dually connected non-orthogonal multiple access based LIFI-RF networks. *Wireless Netw* **31**, 1515–1528 (2025). <https://doi.org/10.1007/s11276-024-03858-3>

54. J. Li, J. Hwang and J. Park, "Indoor Positioning Using Wi-Fi RSSI Based on LSTM-XGBoots Combined Model," 2025 Sixteenth International Conference on Ubiquitous and Future Networks (ICUFN), Lisbon, Portugal, 2025, pp. 532-536, doi: 10.1109/ICUFN65838.2025.11170030.

55. E. Ancillotti, L. Pescosolido and A. Passarella, "Toward Hybrid COTS-based LiFi/WiFi Networks with QoS Requirements in Mobile Environments," 2025 International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), Barcelona, Spain, 2025, pp. 183-190, doi: 10.1109/MSWiM67937.2025.11309163.

56. Adi, P. D. P., Stekelorom, K., Dogheche, E., Armi, N., Remiens, D., & Dayoub, I. (2025). Comprehensive Analysis of Hybrid Approaches to Li-Fi and LPWAN Systems for IoT with Satellite-Assisted Communication. *IETE Technical Review*, 42(5), 597–631. <https://doi.org/10.1080/02564602.2025.2560812>

57. Ali, W.A., Shaker, M.H., Shokair, M. *et al.* Green underwater communication for ROVs: harnessing Li-Fi for sustainable and efficient marine operations. *J Opt* (2025). <https://doi.org/10.1007/s12596-025-02774-9>

58. Chatterjee, B., Rajan, S., Kumar, A. *et al.* Simulation and fabrication of underwater communication system through Li-Fi technology at different scenarios. *J Opt* (2025). <https://doi.org/10.1007/s12596-025-02863-9>

59. Jeffin Gracewell, Sirivelu Veddepya, Swetha Jeevith; Case study on implementation of Li-Fi technologies in controlling home appliances using IoT. *AIP Conf. Proc.* 9 January 2025; 3159 (1): 030011. <https://doi.org/10.1063/5.0247319>

60. E. Mahalal, E. Hasan, M. Ismail, Z. -Y. Wu, M. M. Fouda and Z. M. Fadlullah, "Deep Learning-based Physical Layer Authentication in LiFi Networks Under Multi - User Mobility," SoutheastCon 2025, Concord, NC, USA, 2025, pp. 776-781, doi: 10.1109/SoutheastCon56624.2025.10971591

61. Li, C., Du, X., Hou, X. *et al.* An indoor positioning optimization method based on dimensionality reduction and clustering fusion. *J. King Saud Univ. Comput. Inf. Sci.* **37**, 118 (2025). <https://doi.org/10.1007/s44443-025-00132-0>

62. Ranjan, Rajeev and Sharma, Pratham Vinayak. "Simulation-based modeling of quantum key distribution in a Li-Fi system using free-space optics and polarization modulation" *Journal of Optical Communications.* <https://doi.org/10.1515/joc-2025-0235>

63. Razaque, A., Bektemyssova, G., Khan, M. *et al.* Comprehensive analysis of Li-Fi technology: positioning algorithms, security vulnerabilities, and future IoT applications. *Telecommun Syst* **88**, 130 (2025). <https://doi.org/10.1007/s11235-025-01358-z>

64. Hailu, Tesfay Gidey, Xiansheng Guo, and Haonan Si. 2025. "Indoor Positioning Systems as Critical Infrastructure: An Assessment for Enhanced Location-Based Services" *Sensors* 25, no. 16: 4914. <https://doi.org/10.3390/s25164914>

65. L. Yang, L. Li, J. Wei, P. Cao, Y. Wang and W. Wang, "Large-Scale Omnidirectional Person Positioning," in IEEE Transactions on Pattern Analysis and

Machine Intelligence, vol. 48, no. 3, pp. 2870-2885, March 2026, doi: 10.1109/TPAMI.2025.3630317.

66. M. Wasim Abbas Ashraf, S. Shrivastava, O. Jee Pandey, A. R. Singh and A. Raza, "DRL-Driven Optimal User Association and Load Balancing in Hybrid RF/LiFi-Based IoT Systems," in IEEE Transactions on Network and Service Management, vol. 22, no. 6, pp. 5503-5516, Dec. 2025, doi: 10.1109/TNSM.2025.3607433.

67. Kourdi, Z., Adardour, H.E., Four, I. *et al.* Improved AlGaN–GaN MQW LED performance incorporating InAlN MLB and AlN back barrier: Li-Fi applications. *Indian J Phys* **99**, 651–660 (2025). <https://doi.org/10.1007/s12648-024-03281-6>

68. J. Rajasubha, A. Sathish, B. Jackson, U. Hemavathi, A. P. Ajay and G. Ali, "Next-Gen Smart Homes: AI-Enhanced Li-Fi for Superior Efficiency and Protection," 2025 6th International Conference for Emerging Technology (INCET), BELGAUM, India, 2025, pp. 1-6, doi: 10.1109/INCET64471.2025.11140096.

69. Rao, G.S., Ramadevi, A., Krishna, P.S.R. (2025). Smart Mobility and Health Monitoring System: Revolutionizing Wheelchair and Bed. In: Kumar, S., Mary Anita, E.A., Kim, J.H., Nagar, A. (eds) Fifth Congress on Intelligent Systems. CIS 2024. Lecture Notes in Networks and Systems, vol 1276. Springer, Singapore. https://doi.org/10.1007/978-981-96-2697-7_25

70. Abdullah Khawatmi, Nasir Saeed, Mohamed Atef. 4-PAM Visible Light Communication System with Binary Neural Tree Equalization. *Adv Devices Instrum.* 0:DOI:10.34133/adi.0177

71. Мікляве Г.О. Рішення задачі позиціонування всередині приміщень на базі комплексного використання методів Fingerprint та RSS, Всеукраїнський конкурс студентських наукових робіт з галузей знань і спеціальностей у 2025/2026 н.р.

72. Романов, О. І. ., & Мікляєв, Г. О. . (2025). ПРОГНОЗУВАННЯ СПОТВОРЕНИХ СИГНАЛІВ ВІД LED ЛАМПІ В СИСТЕМАХ ПОЗИЦІОНУВАННЯ ВСЕРЕДИНІ ПРИМІЩЕНЬ. *Збірник матеріалів Міжнародної науково-технічної конференції «ПЕРСПЕКТИВИ ТЕЛЕКОМУНІКАЦІЙ»*, 120–122. вилучено із <https://conferenc-journal.its.kpi.ua/article/view/331874>

73. Romanov, O., & Miklaiv, G. (2021). THROUGHPUT INCREASE IN THE OFFICE LI-FI SYSTEM. *Information and Telecommunication Sciences*, 12(2), 16–21. <https://doi.org/10.20535/2411-2976.22021.16-21>

ДОДАТОК А

ЛІСТИНГ ПРОГРАМИ

А.1 Лістинг основного коду

```

from os import rename

import numpy as np
import yaml
import pandas as pd
import matplotlib.pyplot as plt
from fontTools.subset import subset
from scipy.spatial.distance import cdist

CUTOFF = 2e-6
CUTOFF_HARD = 1e-7
CUTOFF_EXTREME = 1e-10

def calculation():
    with open('config.yml', 'r') as ymlfile:
        cfg = yaml.safe_load(ymlfile)

    fov_tx = cfg['fov_tx'] * np.pi / 180
    fov_rx = cfg['fov_rx'] * np.pi / 180
    n = cfg['n']
    s = cfg['S']
    t = cfg['T']
    l = np.array(cfg['L'])
    v = np.array(cfg['V'])
    step = cfg['step']
    sample_rate = cfg['sample_rate']
    p_tr = cfg['P_tr']
    gfov = (n / np.sin(fov_rx)) ** 2
    m = -np.log(2) / np.log(np.cos(fov_tx))
    lamp_count = l.shape[0]
    left_shift = step / 2
    right_shift = -step / 2
    xmas = np.arange(left_shift, v[0] + right_shift + step, step).round(3)
    ymas = np.arange(left_shift, v[1] + right_shift + step, step).round(3)
    df_lamp = pd.DataFrame(l, columns=['x_lamp', 'y_lamp', 'z_lamp'])
    df_lamp['lamp'] = df_lamp.index
    lamp_mas = np.arange(lamp_count)
    x, y, lamps = np.meshgrid(xmas, ymas, lamp_mas)
    df_coordinates = pd.DataFrame({'x_fingerprint': x.flatten(), 'y_fingerprint': y.flatten(),
                                  'lamp': lamps.flatten()})
    df_coordinates['z_fingerprint'] = 0
    df = pd.merge(df_lamp, df_coordinates, on='lamp')
    df['distance'] = np.sqrt((df['x_fingerprint']-df['x_lamp'])**2 + (df['y_fingerprint']-df['y_lamp'])**2 +
                             (df['z_fingerprint']-df['z_lamp'])**2)
    df['power_fingerprint'] = (p_tr * s * (m + 1) * t * gfov * (df['z_lamp'] / df['distance']) ** m) / (2 * np.pi * df['distance']
    ** 2)
    df['power_fingerprint_int'] = df['power_fingerprint'] / sample_rate
    df['power_fingerprint_int'] = df['power_fingerprint_int'].astype(np.uint16)
    df.to_csv('fingerprint_power.csv', index=False)

def distort_user_signal(df_undistorted, distortion_coefficient, distorted_lamps):

```

```

df_distorted = df_undistorted.rename(columns={'power_user': 'power_user_undistorted',
                                             'power_user_int': 'power_user_undistorted_int'})
df_distorted['power_user_distorted'] = df_distorted.apply(lambda row: row['power_user_undistorted'] *
distortion_coefficient if row['lamp'] in distorted_lamps else row['power_user_undistorted'], axis=1)
# df_distorted['power_user_distorted'] = df_distorted['power_user_undistorted']
# df_distorted['power_user_distorted'] =
df_distorted[df_distorted['lamp'].isin(distorted_lamps)]['power_user_distorted'] * distortion_coefficient
return df_distorted

def extract_user_signal(df_fingerprint,x,y):
df_user = df_fingerprint.loc[((df_fingerprint['x_fingerprint'] == x) & (df_fingerprint['y_fingerprint'] == y))]
df_user = df_user.reset_index(drop=True)
df_user = df_user.rename(columns={'x_fingerprint': 'x_user_real', 'y_fingerprint': 'y_user_real',
                                'z_fingerprint': 'z_user_real', 'power_fingerprint': 'power_user',
                                'power_fingerprint_int': 'power_user_int'})

return df_user

def calculate_error(df_fingerprint, df_user):
df = pd.merge(df_fingerprint, df_user, on='lamp')
df['diff'] = df['power_fingerprint'] - df['power_user_distorted']
return df

# def draw_circle(df, lamp, circle)
def draw_plot():
    with open('config.yml', 'r') as ymlfile:
        cfg = yaml.safe_load(ymlfile)
    L = np.array(cfg['L'])
    x = L[:,0]
    y = L[:,1]
    plt.figure(figsize=(5, 5))
    plt.xlim(0, 4)
    plt.ylim(0, 4)
    small_range = np.arange(0, 4.1, 0.1)
    big_range = np.arange(0, 4.5, 0.5)
    plt.grid(True)
    axes = plt.gca()
    axes.set_xticks(big_range)
    axes.set_yticks(big_range)
    axes.set_xticks(small_range, minor=True)
    axes.set_yticks(small_range, minor=True)
    axes.grid(which='minor', color='black', linestyle='-', linewidth=0.5, alpha=0.3)
    axes.grid(which='major', color='black', linestyle='-', linewidth=1, alpha=0.6)
    plt.scatter(x,y)
    # plt.show()

def plot_best_suit_circle(df):
    draw_plot()
    cond = np.abs(df['diff']) < 1e-6
    df_filtered = df[cond]
    for color, label in zip('gbmr', [0, 1, 2, 3]):
        subset = df_filtered[df_filtered.lamp == label]
        plt.scatter(subset.x_fingerprint, subset.y_fingerprint, s=10, c=color)
    plt.show()

def plot_best_suit_point(df):
    draw_plot()
    lamp_combos = df['lamp_combo'].unique()
    for lamp_combo in lamp_combos:
        subset = df[df.lamp_combo == lamp_combo]

```

```

plt.scatter(subset.x_fingerprint, subset.y_fingerprint, s=10)
plt.show()

def root_mean_square_two_elements(a,b):
    return np.sqrt(a**2+b**2)

def df_applying(x):
    df = pd.DataFrame(columns=['lamp_combo','diff_rms'])
    x_reseted = x.reset_index()
    combo_list = ()
    if len(x_reseted) == 2:
        combo_list = ((0,1),)
    if len(x_reseted) == 3:
        combo_list = ((0,1),(0,2),(1,2))
    if len(x_reseted) == 4:
        combo_list = ((0,1),(0,2),(0,3),(1,2),(1,3),(2,3))
    for combo in combo_list:
        diff_rms = root_mean_square_two_elements(x_reseted['diff'][combo[0]], x_reseted['diff'][combo[1]])
        lamp_combo = (x_reseted['lamp'][combo[0]].item(), x_reseted['lamp'][combo[1]].item())
        df.loc[len(df)] = [lamp_combo, diff_rms]
    return df
# return pd.Series(df, index=['lamp_combo', 'diff_sum'])

def root_mean_square(series):
    """Calculates the root mean square of a pandas Series."""
    return np.sqrt(np.mean(series**2))

def tuple_function(series):
    """Returns a tuple of the unique values in a pandas Series."""
    return tuple(series)

def select_best_suit(df):
    cond = np.abs(df['diff']) < CUTOFF
    df_filtered = df[cond]
    df_intersection = df_filtered[df_filtered.duplicated(subset=['x_fingerprint', 'y_fingerprint'], keep=False)]
    df_merged = df_intersection.groupby(['x_fingerprint', 'y_fingerprint']).apply(df_applying).reset_index() #.agg({'diff':
root_mean_square, 'lamp': tuple_function}).reset_index()
#
lamp_combos = df_merged['lamp_combo'].unique()
cut_cond = df_merged['diff_rms'] <= CUTOFF_EXTREME
for lamp_combo in lamp_combos:
    df_subset = df_merged[df_merged['lamp_combo'] == lamp_combo]
    min_diff = df_subset['diff_rms'].min()
    cut_cond = cut_cond | (df_subset['diff_rms'] <= min_diff*1.1)
df_cutted = df_merged[cut_cond]
df_the_best = df_cutted[df_cutted['diff_rms'].abs() <= CUTOFF_EXTREME]
best_lamp_combo = df_the_best.iloc[0]['lamp_combo'] #Add a check, is the lamp combo unique
df_merged_other_lamps = df_cutted[df_cutted['lamp_combo'] != best_lamp_combo]
df_merged_other_lamps['auxiliary'] = 1
df_the_best['auxiliary'] = 0
df_result = pd.concat([df_the_best, df_merged_other_lamps], ignore_index=True)
# print(df_merged.dtypes)
return df_result, best_lamp_combo

def find_the_best(df, lamp_combo):

```

```

df_filtered = df[df['lamp'].isin(lamp_combo)]
aggregated_df = df_filtered.groupby(['x_fingerprint', 'y_fingerprint'])['diff'].agg(root_mean_square).reset_index()
min_val = aggregated_df[aggregated_df['diff'].imin()].reset_index()
min_val['auxiliary'] = 2
return min_val

```

```

def calculate_distances_with_opposite_flags(df, auxiliary):

```

```

    """

```

```

    Calculates the distances between points with opposite flags in a DataFrame.

```

```

    Args:

```

```

    df: A pandas DataFrame with columns 'x', 'y', and 'auxiliary'.

```

```

    Returns:

```

```

    A new pandas DataFrame with columns 'x1', 'y1', 'x2', 'y2', 'distance'.

```

```

    """

```

```

# if not all(col in df.columns for col in ['x', 'y', 'flag']):

```

```

#     raise ValueError("DataFrame must contain 'x', 'y', and 'flag' columns.")

```

```

# Separate points based on their flag

```

```

flag1_points = df[df['auxiliary'] == 0][['x_fingerprint', 'y_fingerprint']].values

```

```

flag2_points = df[df['auxiliary'].isin(auxiliary)][['x_fingerprint', 'y_fingerprint']].values

```

```

flag1_df = df[df['auxiliary'] == 0][['x_fingerprint', 'y_fingerprint']]

```

```

flag2_df = df[df['auxiliary'].isin(auxiliary)][['x_fingerprint', 'y_fingerprint']]

```

```

# Create a list of tuples containing the point coordinates and distances

```

```

results = []

```

```

if len(flag2_points) == 0:

```

```

    if len(flag1_points) == 1:

```

```

        results.append({

```

```

            'x1_fingerprint': flag1_df.iloc[0]['x_fingerprint'],

```

```

            'y1_fingerprint': flag1_df.iloc[0]['y_fingerprint'],

```

```

            'x2_fingerprint': flag1_df.iloc[0]['x_fingerprint'],

```

```

            'y2_fingerprint': flag1_df.iloc[0]['y_fingerprint'],

```

```

            'distance': 0

```

```

        })

```

```

    result_df = pd.DataFrame(results)

```

```

    return result_df

```

```

else:

```

```

    result_df = pd.DataFrame(results)

```

```

    return result_df

```

```

# Calculate distances using cdist

```

```

distances = cdist(flag1_points, flag2_points)

```

```

for i in range(distances.shape[0]):

```

```

    for j in range(distances.shape[1]):

```

```

        results.append({

```

```

            'x1_fingerprint': flag1_df.iloc[i]['x_fingerprint'],

```

```

            'y1_fingerprint': flag1_df.iloc[i]['y_fingerprint'],

```

```

            'x2_fingerprint': flag2_df.iloc[j]['x_fingerprint'],

```

```

            'y2_fingerprint': flag2_df.iloc[j]['y_fingerprint'],

```

```

            'distance': distances[i, j]

```

```

        })

```

```

# Create a DataFrame from the results
result_df = pd.DataFrame(results)
return result_df

def find_minimal_distance(df):
    if df.empty:
        return [2,2]
    df_grouped = df.groupby(['x2_fingerprint', 'y2_fingerprint'])
    df_minimal_value = df[df['distance'] <= df_grouped['distance'].transform('min') * 1.001]
    df_counted = df_minimal_value.value_counts(subset=['x1_fingerprint', 'y1_fingerprint'])
    #df_minimal_value.groupby(['x1_fingerprint', 'y1_fingerprint']).size()
    max_value = df_counted.max()
    is_max_unique = (df_counted == max_value).sum() == 1
    if is_max_unique:
        return df_counted.idxmax()
    else:
        return [2,2]
    # result = df_counted.idxmax()
    # return result

def find_location(df_user, df_fingerprint, save_result):
    df_with_error = calculate_error(df_fingerprint, df_user)
    df_lamp_combos, lamp_combo = select_best_suit(df_with_error)
    aux_point = find_the_best(df_with_error, [0,1,2,3])
    df_lamp_combos = pd.concat([df_lamp_combos, aux_point], ignore_index=True)
    df_distances = calculate_distances_with_opposite_flags(df_lamp_combos, 2)
    if save_result:
        df_with_error.to_csv('df_with_error.csv', index=False)
        df_lamp_combos.to_csv('df_best_suit.csv', index=False)
        df_distances.to_csv('df_distances.csv', index=False)
        plot_best_suit_circle(df_with_error)
        plot_best_suit_point(df_lamp_combos)
    one_coordinate = find_minimal_distance(df_distances)
    return one_coordinate

def plot_results(df):
    draw_plot()
    true_subset = df[df.result == True]
    plt.scatter(true_subset.x, true_subset.y, s=10, c='g')
    false_subset = df[df.result == False]
    plt.scatter(false_subset.x, false_subset.y, s=10, c='r')
    plt.show()

def main():
    # calculation()
    with open('config.yml', 'r') as ymlfile:
        cfg = yaml.safe_load(ymlfile)
    distortion_coefficient = cfg['distortion_coefficient']
    distorted_lamps = cfg['distorted_lamps']
    df_fingerprint = pd.read_csv('fingerprint_power.csv')
    all_coords = cfg['process_all_coords']
    if all_coords:
        df_results = pd.DataFrame(columns=['x', 'y', 'result'])
        for input_x100 in range(5, 405, 10):
            input_x = input_x100/100
            for input_y100 in range(5, 405, 10):
                input_y = input_y100 / 100
                df_user = extract_user_signal(df_fingerprint, x=input_x, y=input_y)

```

```

df_distorted = distort_user_signal(df_user, distortion_coefficient, distorted_lamps)
calculated_coordinate = find_location(df_distorted, df_fingerprint, False)
if (calculated_coordinate[0] == input_x) and (calculated_coordinate[1] == input_y):
    df_results.loc[len(df_results)] = [input_x, input_y, True]
else:
    df_results.loc[len(df_results)] = [input_x, input_y, False]
plot_results(df_results)
df_results.to_csv('df_results.csv', index=False)
else:
    input_x = cfg['input_coordinate'][0]
    input_y = cfg['input_coordinate'][1]
    df_user = extract_user_signal(df_fingerprint, input_x, input_y)
    df_distorted = distort_user_signal(df_user, distortion_coefficient, distorted_lamps)
    calculated_coordinate = find_location(df_distorted, df_fingerprint, True)
    print(calculated_coordinate)
#
# step = cfg['step']
#
#
# print(one_coordinate)

if __name__ == '__main__':
    main()

```

A.2 Лістинг налаштувань конфігурації

```

n: 1.5
S: 0.0001
fov_tx: 70
T: 1
fov_rx: 90
P_tr: 10
L:
- [1.5, 1.5, 3]
- [1.5, 2.5, 3]
- [2.5, 1.5, 3]
- [2.5, 2.5, 3]
V: [4, 4, 3]
step: 0.1
sample_rate: 1.0e-8
input_coordinate: [0.05, 0.15]
input_error_lamp: 4
distortion_coefficient: 1.2
input_error_coefficients:
- 0.2
- 0.4
- 0.6
- 0.8
- 1.2
- 1.4
- 1.6
- 1.8
Lamp_combos_m2:
- [1, 2]
- [1, 3]
- [1, 4]
- [2, 3]
- [2, 4]
- [3, 4]

```

```
Lamp_combos_m1:  
- [1, 2, 3]  
- [1, 2, 4]  
- [1, 3, 4]  
- [2, 3, 4]  
distorted_lamps: [2,3]  
process_all_coords: True
```

ДОДАТОК Б

ПОЧАТКОВІ ПАРАМЕТРИ

Таблиця Б.1

Параметри системи Li-Fi

Назва показника	Умовне позначення	Одиниці виміру	Значення
<i>Показники приймача</i>			
Показник заломлення	n		1,5
Площа покриття фотоприймача	S	м ²	0,0001
Максимальний кут приймання	FOV _{Rx}	°	90
Коефіцієнт підсилення оптичного фільтра	g		1
<i>Показники передавача</i>			
Потужність передавання	P _{Tx}	Вт	10
Максимальний кут передавання	FOV _{Tx}	°	70
Коефіцієнт підсилення оптичного фільтра	T		1
<i>Показники кімнати</i>			
Довжина кімнати	L	м	4
Ширина кімнати	W	м	4
Висота кімнати	H	м	3
<i>Показники розташування ламп</i>			
Лампа 1	L ₁	м	(1;1)
Лампа 2	L ₂	м	(1;3)
Лампа 3	L ₃	м	(3;1)
Лампа 4	L ₄	м	(3;3)
<i>Показники при розрахунку</i>			
Крок між опорними точками	step	м	0,1