ДОДАТОК А

графічний матеріал кваліфікаційної роботи

Харківський національний університет радіоелектроніки
Кафедра ЕОМ

МЕТОДИ МОНІТОРИНГУ ВУЗЛІВ КОМП'ЮТЕРНОЇ МЕРЕЖІ НА БАЗІ РОЮ БПЛА

КВАЛІФІКАЦІЙНА РОБОТА
ДРУГИЙ (МАГІСТЕРСЬКИЙ) РІВЕНЬ

АВТОР:
ПОГОРЕЛЕНКО В.В.,
СТУД. ГР. СПМ22-2

КЕРІВНИК:
ТКАЧОВ В.М.,
ДОЦ. КАФ. ЕОМ

Мета і задачі роботи

2

МЕТА: ДОСЛІДЖЕННЯ ТА РОЗРОБКА МЕТОДІВ МОНІТОРИНГУ ВУЗЛІВ КОМП'ЮТЕРНОЇ МЕРЕЖІ НА ОСНОВІ ВИКОРИСТАННЯ РОЮ БЕЗПІЛОТНИХ ЛІТАЛЬНИХ АПАРАТІВ (БПЛА) З МЕТОЮ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ , ПРОПУСКНОЇ СПРОМОЖНОСТІ ТА НАДІЙНОСТІ ЗВ'ЯЗКУ.

ЗАДАЧІ:

- ДОСЛІДЖЕННЯ ТЕХНОЛОГІЇ ПЕРЕДАЧІ ДАНИХ У ВИСКОМОБІЛЬНИХ МЕРЕЖАХ
- ДОСЛІДЖЕННЯ ВІДОМИХ РІШЕНЬ В ДАНІЙ ПРЕДМЕТНІЙ ОБЛАСТІ
- РОЗРОБКА МЕТОДІВ МОНІТОРИНГУ ВУЗЛІВ ВИСОКОМОБІЛЬНОЇ МЕРЕЖІ
- АПРОБАЦІЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ
- ФОРМУЛЮВАННЯ РЕКОМЕНДАЦІЙ, ЩОДО ПРАКТИЧНОГО ВПРОВАДЖДЕННЯ РОЗРОБЛЕНИХ МЕТОДІВ

3

Предметна область

- МЕТОДИМОНІТОРИНГУКОМП'ЮТЕРНОЇ МЕРЕЖІ
- ОСНОВНІ ХАРАКТЕРИСТИКИ
- ОСНОВНІ ВИКЛИКИ

4

Огляд відомих рішень

Метод	Опис	Переваги	Обмеження
Аналіз пропускної спроможності (Bandwidth Analysis)	Вимірювання та аналіз швидкості передачі даних для визначення пропускної спроможності та ідентифікації заторів.	Об'єктивність, виявлення заторів	Вимагає спеціалізованого обладнання
SNMP (Simple Network Management Protocol)	Протокол для моніторингу та управління мережевими пристроями.	Збір інформації про стан обладнання	Не завжди надає детальну аналітику
Моніторинг використання ресурсів (Resource Utilization Monitoring)	Відстежування використання ресурсів (процесор, пам'ять, сховище) для виявлення аномалій.	Виявлення неефективного використання ресурсів	Фокусується на ресурсах, а не на мережі

Технології, що засновані на використанні методу Bandwidth Analysis

5

Прямий RF-зв'язок:

Прямий RF-зв'язок використовує частотну модуляцію для передачі даних. Цей метод є ефективним для коротких дистанцій і простий у встановленні, забезпечуючи надійне з'єднання з мінімальними затратами.

Меш-мережі:

Меш-мережі використовують множину вузлів для створення гнучкої та розподіленої мережі. Це дозволяє покривати великі території та підтримувати з'єднання навіть при виході з ладу окремих вузлів.

Супутниковий зв'язок:

Супутниковий зв'язок використовує орбітальні супутники для передачі даних, забезпечуючи глобальне покриття та високу дальність зв'язку.

Критерії	Прямий RF-зв'язок	Меш-мережі	Супутниковий зв'язок
Використання	Частотна модуляція для передачі даних	Використання множини вузлів для створення мережі	Використання орбітальних супутників для передачі даних
Ефективність	Ефективний для коротких дистанцій	Дозволяє покривати великі території	Забезпечує глобальне покриття
Складність встановлення	Низька складність	Середня складність через необхідність конфігурації мережі	Висока складність, потребує спеціалізованого обладнання
Надійність з'єднання	Надійне з'єднання	З'єднання не втрачається навіть при виході з ладу вузлів	Висока надійність, але на зв'язок впливають погодні умови
Покриття	Обмежене короткими відстанями	Гнучке та розподільне покриття	Висока дальність зв'язку

Адаптивний Шенон-базований метод передачі даних (ASHBMT)

6

- **Адаптивний** : підкреслює здатність методу адаптуватися до змінних умов.
- **Шенон-базований**: вказує на використання основних принципів формули Шенона.
- **Метод передачі даних**: акцентує на основній функції методу

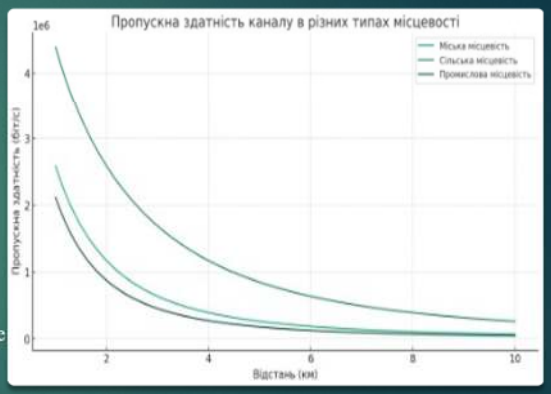
7

Адаптивний Шенон-базований метод передачі даних (ASHBMT)

$$C = B \cdot log_2\left(1 + \frac{P \cdot G(d)}{N}\right)$$

де:

- C — пропускна здатність каналу в бітах за секунду.
- B — смуга пропускання каналу в герцах.
- P — потужність сигналу ЗАДАТИ ОДИНИЦІ (яка може адаптуватися в залежності від відстані та інших умов).
- $G(d)$ — функція залежності ефективності передачі сигналу від відстані d між передавачем та приймачем ЗАДАТИ ОДИНИЦІ.
- N — потужність шуму дБ.



8

Сукупність технічних вимог для постановки експерименту

- **Високоточне обладнання для вимірювання відстані**
- **Обладнання для моніторингу та аналізу сигналу**
- **Система керування потужністю сигналу**
- **Модуль обробки сигналу для реалізації формули Шенона**
- **Система MIMO (Multiple Input Multiple Output)**
- **Адаптивне програмне забезпечення**
- **Технічна підтримка та обслуговування**

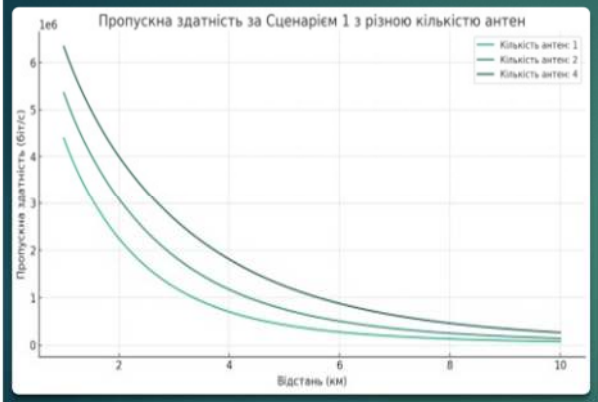
Модельний експеримент

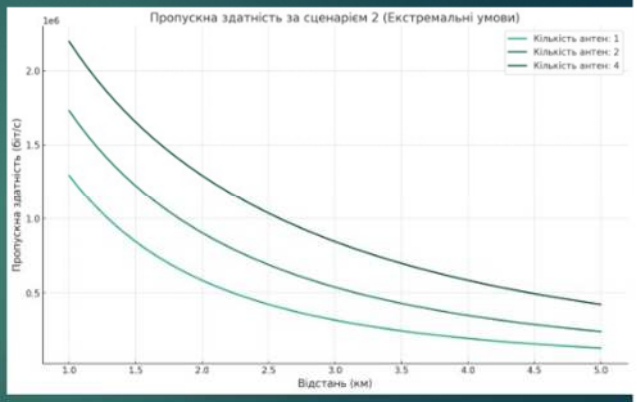
9

Відстань(км)	Пропускна здатність 1 антена (біт/с)	Пропускна здатність 2 антена (біт/с)	Пропускна здатність 4 антена (біт/с)
1.0	3.46e+06	4.39e+06	5.36e+06
1.09	3.46e+06	4.39e+06	5.36e+06
….	….	….	….
9.91	3.46e+06	4.39e+06	5.36e+06
10.0	3.46e+06	4.39e+06	5.36e+06

Результати експерименту

10





Апробація Результатів Дослідження

11

ВАЖЛИВІСТЬ ОПТИМІЗАЦІЇ АЛГОРИТМІВ УПРАВЛІННЯ РОЄМ БПЛА ДЛЯ ЕФЕКТИВНОГО МОНІТОРИНГУ МЕРЕЖІ

09.11.2023 12:14

[1. Інформаційні системи і технології]

Автор: **Погореленко Володимир Володимирович, студент, Харківський національний університет радіоелектроніки, м. Харків**

ORCID: 0009-0009-3003-9524

ПОСИЛАННЯ НА СТАТТЮ: HTTP://WWW.KONFERENCIAONLINE.ORG.UA/UA/ARTICLE/ID-1433/

Перспективи Розвитку та Можливі Застосування

12

РОЗВИТОК ТА ОПТИМІЗАЦІЯ ASHBMT

ПРОДОВЖЕННЯ РОЗВИТКУ ASHBMT ВКЛЮЧАТИМЕ ПОДАЛЬШУ ІНТЕГРАЦІЮ З ПЕРЕДОВИМИ АЛГОРИТМАМИ МАШИННОГО НАВЧАННЯ ТА ШТУЧНОГО ІНТЕЛЕКТУ ЦЕ ЗАБЕЗПЕЧИТЬ ЩЕ КРАЩУ АДАПТИВНІСТЬ ТА АВТОМАТИЗАЦІЮ У ВІДПОВІДЬ НА ЗМІНИ В МЕРЕЖЕВОМУ СЕРЕДОВИЩІ. КРІМ ТОГО, МОЖЛИВЕ ВДОСКОНАЛЕННЯ СИСТЕМИ ІМІТАЦІЇ ДЛЯ ОПТИМІЗАЦІЇ ВИКОРИСТАННЯ АНТЕНТА ЗМЕНШЕННЯ ЕНЕРГОСПОЖИВАННЯ

МОЖЛИВІ ЗАСТОСУВАННЯ

- КРИТИЧНО ВАЖЛИВІ МЕРЕЖІ ЗВ'ЯЗКУ: ASHBMT МОЖЕ БУТИ ЗАСТОСОВАНИЙ У КРИТИЧНО ВАЖЛИВИХ МЕРЕЖАХ, ТАКИХ ЯК ВІЙСЬКОВІ АБО АВАРІЙНІ СЛУЖБИ, ДЕ ВИСОКА НАДІЙНІСТЬ ТА СТАБІЛЬНІСТЬ ЗВ'ЯЗКУ Є КЛЮЧОВИМИ
- **ЕКОЛОГІЧНИЙ МОНІТОРИНГ:** ЗАСТОСУВАННЯ БПЛА ДЛЯ МОНІТОРИНГУ ЕКОЛОГІЧНИХ ПАРАМЕТРІВ, ТАКИХ ЯК ЯКІСТЬ ПОВІТРЯ АБО ЗМІНИ В ПРИРОДНИХ ЛАНДШАФТАХ.
- **СІЛЬСЬКЕ ГОСПОДАРСТВО.** ВИКОРИСТАННЯ БПЛА ДЛЯ МОНІТОРИНГУ СТАНУ СІЛЬСЬКОГОСПОДАРСЬКИХ УГІДЬ, ВИЯВЛЕННЯ ХВОРОБ РОСЛИН ТА ОПТИМІЗАЦІЇ ВИКОРИСТАННЯ РЕСУРСІВ

ПОДАЛЬШІ ДОСЛІДЖЕННЯ

РОЗВИТОК ASHBMT ВІДКРИВАЄ МОЖЛИВОСТІ ДЛЯ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ, ОСОБЛИВО У СФЕРІ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ПЕРЕДАЧІ ДАНИХ У СКЛАДНИХ МЕРЕЖЕВИХ УМОВАХ. ОСОБЛИВА УВАГА МОЖЕ БУТИ ПРИДІЛЕНА ВИВЧЕННЮ ВПЛИВУ ЕКСТРЕМАЛЬНИХ УМОВ, ТАКИХ ЯК ВИСОКИЙ РІВЕНЬ ІНТЕРФЕРЕНЦІЇ АБО ОБМЕЖЕНІ МОЖЛИВОСТІ ДЛЯ ВСТАНОВЛЕННЯ МЕРЕЖЕВИХ ВУЗЛІВ.

Основні Висновки Дослідження

13

- ДОСЛІДЖЕНО АСПЕКТИ РАДІОЗВ'ЯЗКУ В МОНІТОРИНГУ МЕРЕЖІ НА БАЗІ РОЮ БЕЗПІЛОТНИХ ЛІТАЛЬНИХ АПАРАТІВ (БПЛА) З МЕТОЮ ЗРОЗУМІННЯ СУТНОСТІ ПРОБЛЕМИ ТА ВИЗНАЧЕННЯ МОЖЛИВИХ ШЛЯХІВ ЇЇ ВИРІШЕННЯ .
- РОЗРОБЛЕНІ ВЛАСНІ МЕТОДИ , ВИКОРИСТОВУЮЧИ АНАЛІЗ ДАНИХ ТА РОЗРАХУНКИ , ЩО ПРИЗВЕЛО ДО ПОКРАЩЕННЯ ЕФЕКТИВНОСТІ СИСТЕМИ МОНІТОРИНГУ.
- РОЗРОБЛЕНІ МАТЕМАТИЧНІ МОДЕЛІ ТА ПРОВЕДЕНО ТЕОРЕТИЧНЕ МОДЕЛЮВАННЯ , ЩО ПІДТВЕРДИЛО ЙОГО ВІРОГІДНІСТЬ ТА ЕФЕКТИВНІСТЬ .
- ПРОВЕДЕНА АПРОБАЦІЯ МЕТОДУ В ГЕТЕРОГЕННОМУ СЕРЕДОВИЩІ ТА ВИЗНАЧЕНО КОРДОНИ ЗАСТОСУВАННЯ ОБРАНОГО МЕТОДУ.
- СФОРМУЛЬОВАНО РЕКОМЕНДАЦІЇ ДЛЯ ПРАКТИЧНОЇ РЕАЛІЗАЦІЇ ТА ВИЗНАЧЕНО МОЖЛИВІ ПЕРЕШКОДИ НА ЦЬОМУ ШЛЯХУ.

ДОДАТОК Б
ЛІСТИНГ ПРОГРАМ ДЛЯ РОЗРАХУНКУ

Б.1 Програми для розрахунку та візуалізації отриманих даних

Б.1.1 Лістинг коду для генерації амплітудно-модульованого сигналу (АМ) на Python

```
import numpy as np
import matplotlib.pyplot as plt

def generate_am_signal(Ac, m, fc, fm, duration, Fs):
    """
    Генерація амплітудно-модульованого сигналу.

    Параметри:
    Ac (float): Амплітуда несучої хвилі.
    m (float): Індекс модуляції.
    fc (float): Частота несучої хвилі.
    fm (float): Частота модулюючого сигналу.
    duration (float): Тривалість сигналу в секундах.
    Fs (float): Частота дискретизації.

    Повертає:
    t (numpy.ndarray): Часовий вектор.
    AM (numpy.ndarray): Амплітудно-модульований сигнал.
    carrier (numpy.ndarray): Несуча хвиля.
    modulating (numpy.ndarray): Модулюючий сигнал.
    """
    t = np.arange(0, duration, 1/Fs)
    carrier = Ac * np.cos(2 * np.pi * fc * t)
    modulating = 1 + m * np.cos(2 * np.pi * fm * t)
    AM = carrier * modulating
    return t, AM, carrier, modulating

# Визначення параметрів
Ac = 1.0           # Амплітуда несучої хвилі
m = 0.5            # Індекс модуляції
fc = 10e3          # Частота несучої хвилі (10 кГц)
fm = 1e3           # Частота модулюючого сигналу (1 кГц)
duration = 0.002   # Тривалість сигналу (2 мс)
Fs = 100 * fc      # Частота дискретизації

# Генерація сигналу
t, AM, carrier, modulating = generate_am_signal(Ac, m, fc, fm,
duration, Fs)
```

```
# Візуалізація сигналів
plt.figure(figsize=(10, 8))

# Візуалізація несучої хвилі
plt.subplot(3, 1, 1)
plt.plot(t, carrier)
plt.title("Несуча хвиля")
plt.xlabel("Час (с)")
plt.ylabel("Амплітуда")

# Візуалізація модулюючого сигналу
plt.subplot(3, 1, 2)
plt.plot(t, modulating)
plt.title("Модулюючий сигнал")
plt.xlabel("Час (с)")
plt.ylabel("Амплітуда")

# Візуаліза
ція амплітудно-модульованого сигналу plt.subplot(3, 1, 3)
plt.plot(t, AM) plt.title("Амплітудно-модульований сигнал (AM)")
plt.xlabel("Час (с)") plt.ylabel("Амплітуда")

plt.tight_layout() plt.show()
```

Б 1.2 Листинг коду для розрахунку функції FSPL та візуалізації отриманих даних, залежно від відстані та частоти

```
import numpy as np
import matplotlib.pyplot as plt

def calculate_fspl(d, f, c=3e8):
    """
    Обчислення втрат сигналу на вільнопросторовому шляху (FSPL).

    Параметри:
    d (float or numpy.ndarray): Відстань між передавачем і
приймачем в метрах.
    f (float): Частота сигналу в Герцах.
    c (float): Швидкість світла в вакуумі (за замовчуванням 3e8
м/с).

    Повертає:
    FSPL (float or numpy.ndarray): FSPL в децибелах.
    """
    return 20 * np.log10(d) + 20 * np.log10(f) + 20 * np.log10(4
* np.pi / c)

# Візуалізація FSPL для різних відстаней при фіксованій частоті
d = np.linspace(1, 10000, 1000) # від 1 м до 10 км
```

```
f = 2.4e9 # Частота 2.4 ГГц (наприклад, Wi-Fi)

fspl = calculate_fspl(d, f)

plt.figure(figsize=(10, 6))
plt.plot(d, fspl)
plt.title("FSPL в залежності від відстані")
plt.xlabel("Відстань (м)")
plt.ylabel("FSPL (дБ)")
plt.grid(True)
plt.show()

# Візуалізація FSPL для різних частот при фіксованій відстані
frequencies = np.linspace
```

Б 1.3 Лістинг для розрахунку формули Шенона

```
import numpy as np
import matplotlib.pyplot as plt

def calculate_capacity(B, SNR):
    """
    Calculate the channel capacity using Shannon's formula.

    Parameters:
    B (float or numpy.ndarray): Bandwidth of the channel in
Hertz.
    SNR (float or numpy.ndarray): Signal-to-Noise Ratio.

    Returns:
    C (float or numpy.ndarray): Channel capacity in bits per
second.
    """
    return B * np.log2(1 + SNR)

# Visualization for different SNR values with a fixed bandwidth
SNR_values = np.linspace(0, 100, 100)  # SNR from 0 to 100
fixed_bandwidth = 1e6  # 1 MHz

capacity_snr = calculate_capacity(fixed_bandwidth, SNR_values)

plt.figure(figsize=(10, 6))
plt.plot(SNR_values, capacity_snr)
plt.title("Channel Capacity as a Function of SNR")
plt.xlabel("Signal-to-Noise Ratio (SNR)")
plt.ylabel("Channel Capacity (bps)")
plt.grid(True)
plt.show()

# Visualization for different bandwidths with a fixed SNR
```

```
bandwidths = np.linspace(1e6, 100e6, 100)  # Bandwidth from 1
MHz to 100 MHz
fixed_snr = 10  # Fixed SNR

capacity_bandwidth = calculate_capacity(bandwidths, fixed_snr)

plt.figure(figsize=(10, 6))
plt.plot(bandwidths, capacity_bandwidth)
plt.title("Channel Capacity as a Function of Bandwidth")
plt.xlabel("Bandwidth (Hz)")
plt.ylabel("Channel Capacity (bps)")
plt.grid(True)
plt.show()
```

Б 1.4 Лістинг для розрахунку максимально можливої дальності зв'язку з урахуванням втрат на вільнопросторовому шляху (FSPL) і FSPL з додатковим затуханням

```
import numpy as np
import matplotlib.pyplot as plt

def calculate_fspl(Pt, Pr, frequency, c=3e8):
    """
    Обчислення втрат на вільнопросторовому шляху (FSPL) для
заданої відстані.

    Параметри:
    Pt (float): Потужність передавача в dBm.
    Pr (float): Чутливість приймача в dBm.
    frequency (float): Частота сигналу в Hz.
    c (float): Швидкість світла у вакуумі (за замовчуванням 3e8
м/с).

    Повертає:
    distance (float): Максимально можлива дистанція зв'язку в
метрах.
    """
    # Переведення Pt і Pr з dBm в dB
    Pt_dB = Pt - 30
    Pr_dB = Pr - 30

    # Обчислення FSPL в dB
    FSPL_dB = Pt_dB - Pr_dB

    # Переведення FSPL з dB у лінійний масштаб
    FSPL_linear = 10 ** (FSPL_dB / 10)

    # Обчислення дистанції
```

```
    distance = c / (4 * np.pi * frequency) *
np.sqrt(FSPL_linear)
    return distance

def calculate_fspl_with_attenuation(Pt, Pr, frequency,
attenuation_per_km, distance_km, c=3e8):
    """
    Обчислення FSPL з додатковим затуханням для заданої
відстані.

    Параметри:
    Pt (float): Потужність передавача в dBm.
    Pr (float): Чутливість приймача в dBm.
    frequency (float): Частота сигналу в Hz.
    attenuation_per_km (float): Затухан
ня на одиницю відстані в dB/км. distance_km (float): Відстань в
кілометрах. c (float): Швидкість світла у вакуумі (за
замовчуванням 3e8 м/с).
markdown
Повертає:
FSPL з затуханням (float): FSPL з урахуванням додаткового
затухання в dB.
"""
# Обчислення базового FSPL
basic_fspl = calculate_fspl(Pt, Pr, frequency, c)

# Обчислення додаткового затухання
additional_attenuation = attenuation_per_km * distance_km

# Обчислення FSPL з затуханням
fspl_with_attenuation = basic_fspl + additional_attenuation
return fspl_with_attenuation
def plot_fspl_and_attenuation(Pt, Pr, frequency,
attenuation_per_km, max_distance_km): """ Візуалізація FSPL і
FSPL з додатковим затуханням в залежності від відстані.
arduino
Параметри:
Pt (float): Потужність передавача в dBm.
Pr (float): Чутливість приймача в dBm.
frequency (float): Частота сигналу в Hz.
attenuation_per_km (float): Затухання на одиницю відстані в
dB/км.
max_distance_km (float): Максимальна відстань для візуалізації в
кілометрах.
"""
distances = np.linspace(0, max_distance_km, 500)
fspl_values = [calculate_fspl(Pt, Pr, frequency, distance *
1000) for distance in distances]
fspl_attenuation_values = [calculate_fspl_with_attenuation(Pt,
Pr, frequency, attenuation_per_km, distance, distance * 1000)
for distance in distances]
```

```
plt.figure(figsize=(10, 6))
plt.plot(distances, fspl_values, label="FSPL")
plt.plot(distances
, fspl_attenuation_values, label="FSPL з затуханням")
plt.title("FSPL та FSPL з затуханням в залежності від відстані")
plt.xlabel("Відстань (км)") plt.ylabel("FSPL (дБ)") plt.legend()
plt.grid(True) plt.show()
```

Б 1.5 Лістинг розрахунку ємності каналу MIMO за допомогою теореми Шенона

```
import numpy as np
import matplotlib.pyplot as plt

def calculate_mimo_capacity(H, SNR_dB):
    """
    Розрахунок ємності каналу для системи MIMO за допомогою
теореми Шеннона.

    Параметри:
    H (numpy.ndarray): Матриця каналу MIMO.
    SNR_dB (float): Відношення сигнал/шум в децибелах.

    Повертає:
    capacity (float): Ємність каналу в бітах за секунду.
    """
    SNR_linear = 10 ** (SNR_dB / 10)  # Перетворення SNR з дБ у
лінійний масштаб
    eigenvalues, _ = np.linalg.eig(np.dot(H, H.T))  # Розрахунок
власних значень HH^T
    capacity = sum(np.log2(1 + SNR_linear * eigenvalue) for
eigenvalue in eigenvalues)
    return capacity

def assess_signal_stability(num_antennas):
    """
    Оцінка стабільності сигналу в системі MIMO з заданою
кількістю антен.

    Параметри:
    num_antennas (int): Кількість антен в системі MIMO.

    Повертає:
    stability (float): Оцінка стабільності сигналу.
    """
    stability = np.log2(num_antennas)  # Проста модель для
оцінки стабільності
    return stability

# MIMO Channel Matrix
```

```
H = np.array([[1.41, 0.32], [0.78, 1.79]])
SNR_dB = 20  # Signal-to-Noise Ratio in decibels

# Calculate channel capacity for MIMO system
mimo_capacity = calculate_mimo_capacity(H, SNR_dB)

# Assess stability for different numbers of antennas
antenna_range = np.arange(1, 5)
stabilities = [assess_signal_stability(num) for num in
antenna_range]

# Visualization
plt.figure(figsize=(12, 6))

# MIMO Channel Capacity Visualization
plt.subplot(1, 2, 1)
plt.bar(['MIMO Capacity'], [mimo_capacity], color='skyblue')
plt.ylabel('Channel Capacity (bits/sec)')
plt.title('MIMO Channel Capacity')

# Stability Visualization
plt.subplot(1, 2, 2)
plt.plot(antenna_range, stabilities, marker='o', linestyle='-',
color='green')
plt.xlabel('Number of Antennas')
plt.ylabel('Stability (Arbitrary Units)')
plt.title('Signal Stability with Increasing Antennas')
plt.xticks(antenna_range)

plt.tight_layout()
plt.show()
```

Б 2 Результати візуалізації розрахунків

Б 2.1 Візуалізація графіків розрахунку АМ сигналу

Несуча хвиля: Це чиста синусоїда з частотою несучої хвилі.

Модулюючий сигнал: Ще одна синусоїда, але з частотою модулюючого сигналу.

Амплітудно-модульований сигнал (АМ): Кінцевий результат, де видно вплив модулюючого сигналу на амплітуду несучої хвилі.

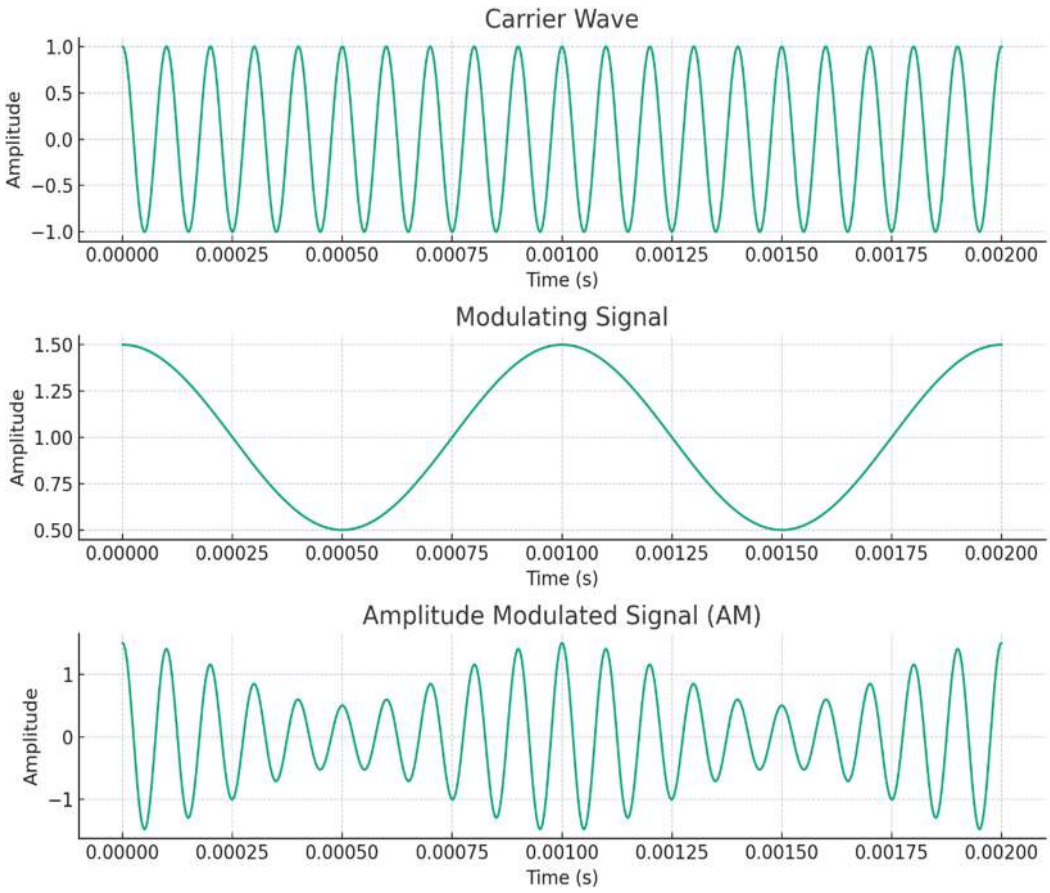


Рисунок 2.1 – Візуалізація результатів розрахунків AM

Б 2.2 Приклад візуалізації отриманих даних за допомогою коду для розрахунку FSPL

1. Візуалізація даних залежно від відстані.

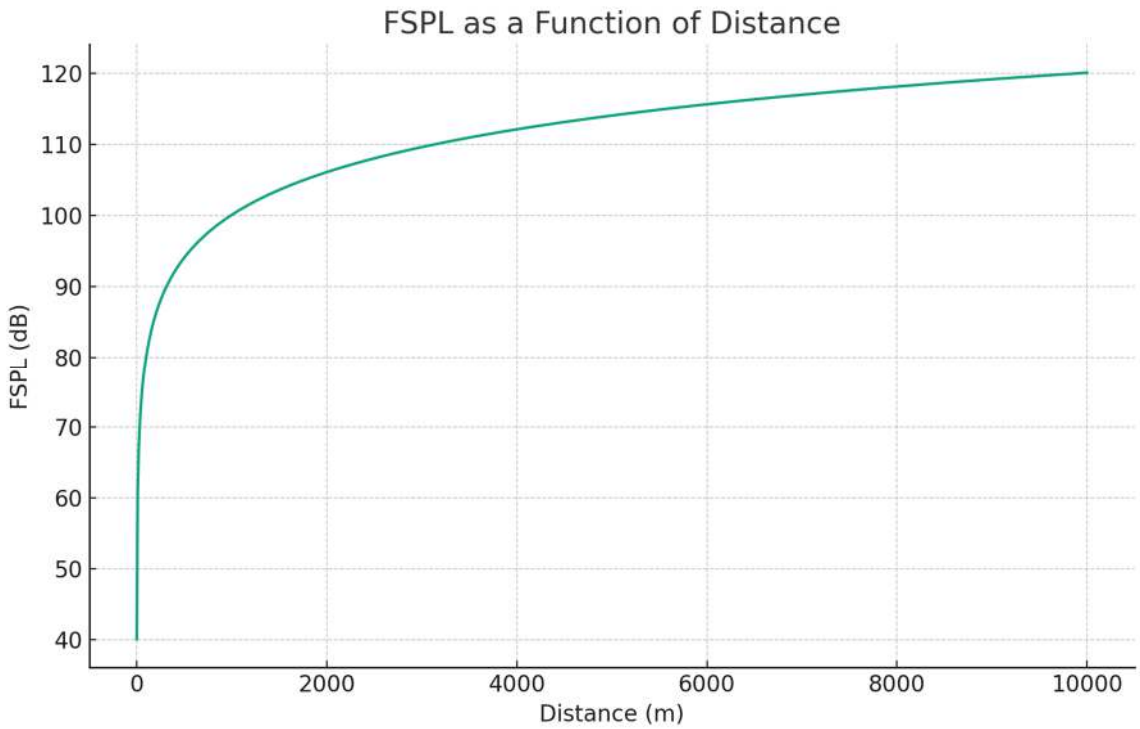


Рисунок 2.2 – Візуалізація отриманих даних залежно від відстані.

2. Візуалізація отриманих даних залежно від частоти.

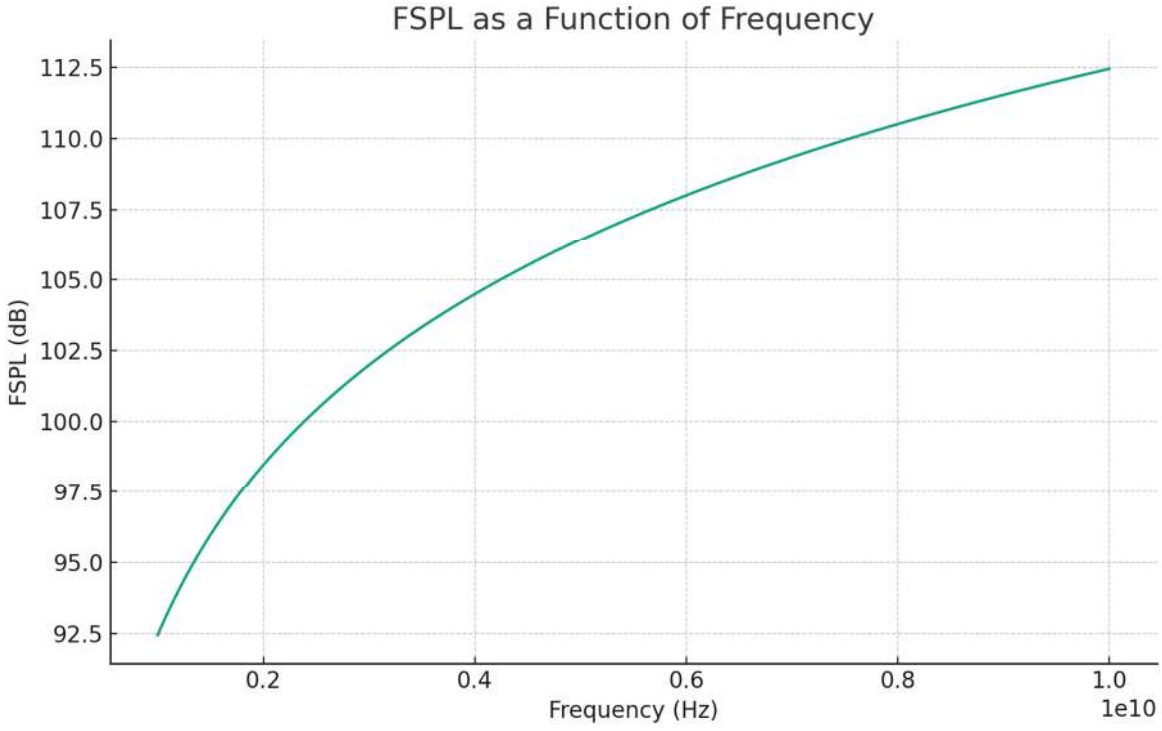


Рисунок 2.3 – Візуалізація отриманих даних залежно від частоти.

Б 2.3 Візуалізація отриманих даних при розрахунку даних за допомогою

формули Шенона

1. Візуалізація того, як максимальна пропускна спроможність каналу змінюється в залежності від відношення сигнал/шум (SNR).

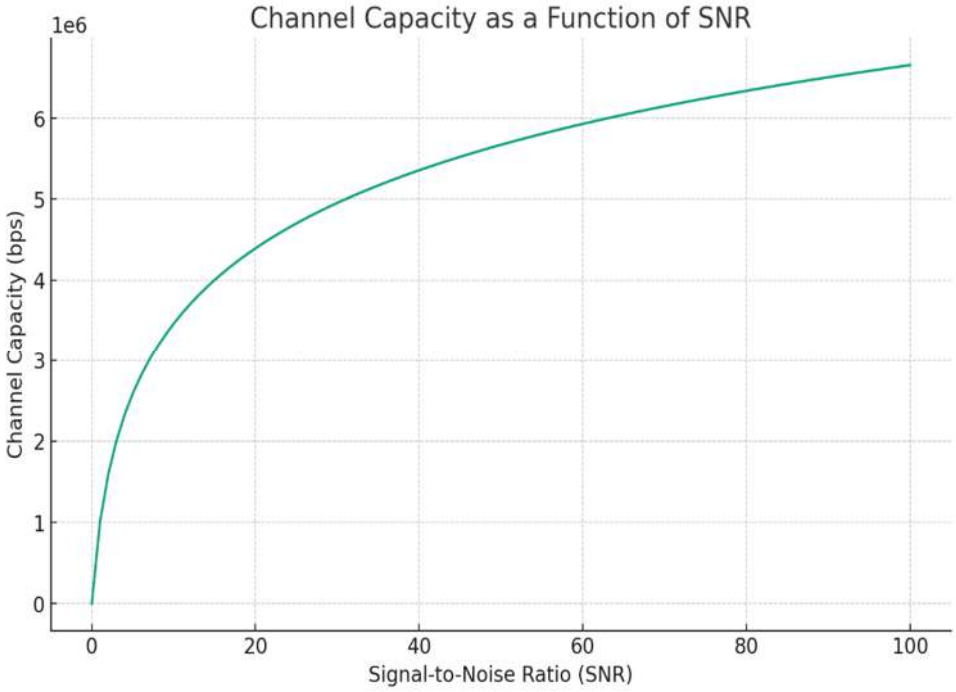


Рисунок 2.3 – Графік відношення сигнал/шум

1. Візуалізація графіку за формулою Шенона.

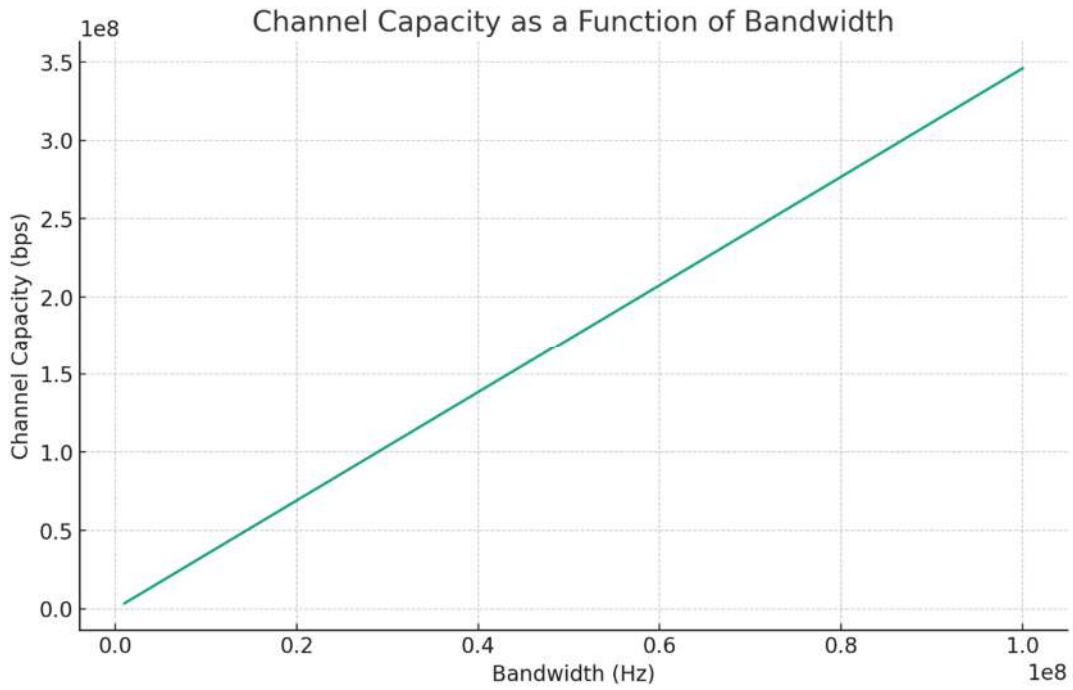


Рисунок 2.4 – Візуалізація графіку за теоремою Шенона

Б 2.4 Візуалізація розрахунку максимально можливої дальності зв'язку з урахуванням втрат на вільнопросторовому шляху (FSPL) і FSPL з додатковим затуханням

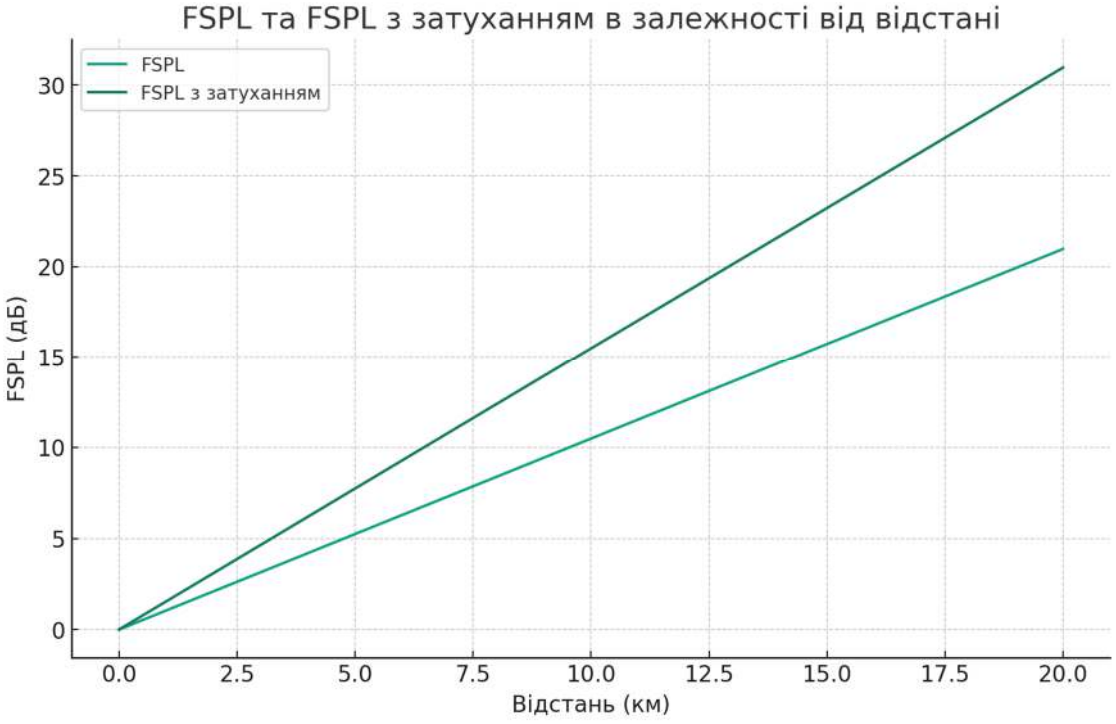


Рисунок 2.5 – Візуалізація отриманих даних

Б 2.5 Візуалізація моделі матриці MIMO

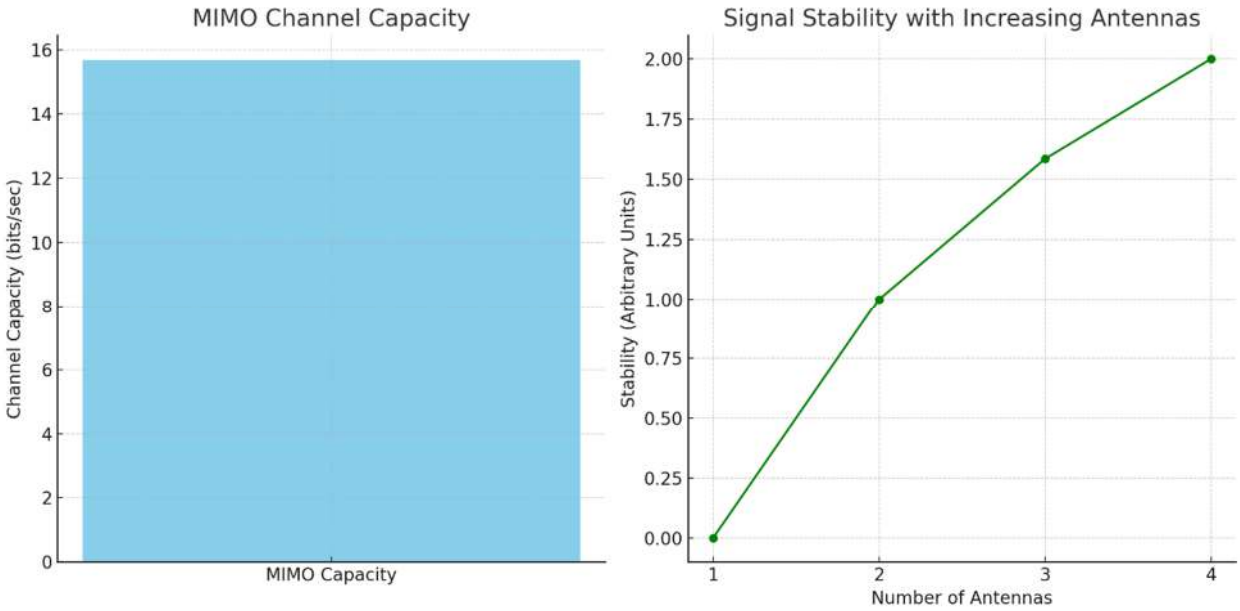


Рисунок 2.6 – Візуалізація отриманих даних за допомогою графіку