

Бакалаврська дипломна робота на тему:

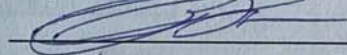
**ПРОГРАМНИЙ ЗАСТОСУНОК ДОСЛІДЖЕННЯ ФОРМАТУ МОДУЛЯЦІЇ В
БЕЗПРОВІДНИХ СИСТЕМАХ ПЕРЕДАЧІ**

Виконав: студент 2-го курсу, групи ПЗТ-22мс
спеціальності 172 – Телекомунікації та
радіотехніка



Мізяківський В. М.

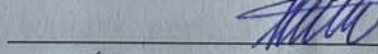
Керівник: к.т.н., доц., доцент каф. ІКСТ



Бортник С.Г.

« 11 » 06 2024 р.

Рецензент: д.т.н., проф., професор каф. ІРТС

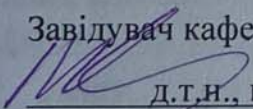


Семенов А.О.

« 11 » 06 2024 р.

Допущено до захисту

Завідувач кафедри ІКСТ



д.т.н., проф. Кичак В.М.

« 11 » 06 2024 р.

Вінницький національний технічний університет
Факультет інформаційних електронних систем
Кафедра інфокомунікаційних систем та технологій
Рівень вищої освіти перший (бакалаврський)
Галузь знань 17 – Електроніка та телекомунікації
(шифр і назва)
Спеціальність 172 – Телекомунікації та радіотехніка
(шифр і назва)
Освітня програма – Програмне забезпечення телекомунікаційних систем

ЗАТВЕРДЖУЮ

Завідувач кафедри ІКСТ

д.т.н., професор В.М. Кичак

«06» 05 2024 року

З А В Д А Н Н Я НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Мізяківському Василю Миколайовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Програмний застосунок дослідження формату модуляції в безпроводних системах передачі

керівник роботи Бортник Сергій Геннадійович, канд. техн. наук, доцент,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом ВНТУ від "11" березня 2024 року № 80

2. Строк подання студентом роботи: 07 червня 2024 року

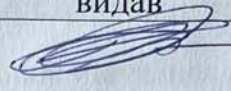
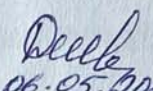
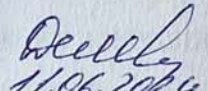
3. Вихідні дані до роботи: тип інфокомунікаційної системи – безпроводна; режим роботи інфокомунікаційної системи - багатоканальна; тип стандарту зв'язку – IEEE 802.11; сумарна швидкість інформаційних потоків - Ethernet 100 Мбіт/с; імовірність помилки в радіотракті - 10^{-6} ; формат модуляції в каналі передавання - багатопозиційна фазова маніпуляція з переходом на комбіновану; типи кодування сигналів – двоетапне із виправленням помилок; метод мультиплексування – частотний з переходом в комбінований; мобільність користувача – до 10 км/год;

площа покриття інфокомунікаційної системи – згідно технічних параметрів телекомунікаційного обладнання Wi-Fi мережі; режим роботи тракту передачі – адаптивний зі зміною формату модуляції, типу кодування та мультиплексування сигналів.

4. Зміст розрахунково-пояснювальної записки роботи (перелік питань, які потрібно розробити): вступ, дослідження форматів модуляції телекомунікаційних сигналів, порівняльний аналіз основних методів модуляції в безпроводних системах передачі даних, оптимізація спектральної щільності потужності модульованих телекомунікаційних сигналів для ефективного використання бездротових систем зв'язку, комп'ютерне моделювання частотної маніпуляції (FSK) радіосигналів в бездротових комунікаційних системах, визначення аспектів забезпечення безпечних та здорових умов праці для всіх працівників.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):
лістинг програми оцінювання PSD модульованих сигналів, дискретна еквівалентна модель для BFSK модуляції, лістинг програми моделювання BFSK телекомунікаційних сигналів, дискретна еквівалентна модель для когерентної демодуляції когерентної FSK, дискретна еквівалентна модель некогерентної демодуляції на основі квадратичного закону для когерентного/некогерентного сигналу FSK, лістинг програми роботи некогерентного демодулятора на основі квадратичного закону.....

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Спеціальна частина	Бортник С.Г., доцент кафедри ІКСТ	 06.05.2024р	 11.06.2024р
Охорона праці	Дембичук С.В. проф. каф. БЖД ПБ	 06.05.2024	 11.06.2024

7. Дата видачі завдання «06» травня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів роботи	Примітка
1	Формування та затвердження теми та індивідуального завдання бакалаврської дипломної роботи (БДР)	06.05.2024р.	
2	Виконання спеціальної частини БДР. Перший рубіжний контроль виконання БДР	20.05.2024р.	
3	Виконання спеціальної частини БДР. Другий рубіжний контроль виконання БДР	31.05.2024р.	
4	Виконання розділу «Охорона праці»	04.06.2024р.	
5	Нормоконтроль БДР	05.06.2024р.	
6	Попередній захист БДР	06.06.2024р.	
7	Рецензування БДР	07.06.2024р.	
8	Захист БДР	12.06.2024р.	

Студент


(підпис)

Мізяківський В.М.

Керівник роботи


(підпис)

Бортник С.Г.

АНОТАЦІЯ

Програмний застосунок дослідження формату модуляції в безпроводних системах передачі. Бакалаврська дипломна робота / В. М. Мізяківський – Вінниця: ВНТУ, 2024. – 105 с., 17 рис., 12 табл., 22 – бібл. – українською мовою.

Метою даної роботи є дослідження форматів модуляції в безпроводних системах передачі для підвищення ефективності передачі даних, адаптацію до змінних умов передачі, розвиток нових технологій, забезпечення безпеки зв'язку та навчання і освіти. Здійснено дослідження форматів модуляції телекомунікаційних сигналів. Виконано порівняльний аналіз основних методів модуляції в безпроводних системах передачі даних. Реалізовано оптимізацію спектральної щільності потужності модульованих телекомунікаційних сигналів для ефективного використання бездротових систем зв'язку. Здійснено комп'ютерне моделювання частотної маніпуляції (FSK) радіосигналів в бездротових комунікаційних системах. Визначено аспекти забезпечення безпечних та здорових умов праці для всіх працівників.

Ключові слова: формат модуляції; безпроводна система передачі; спектральна щільність потужності; комп'ютерне моделювання.

ABSTRACT

Software application of modulation format research in wireless transmission systems. Bachelor's thesis / V. Miziakovsky - Vinnytsia: VNTU, 2024 - 105 p., 17 figs., 12 tables, 22 bibliography - in Ukrainian.

The purpose of this work is to study modulation formats in wireless transmission systems to improve data transmission efficiency, adaptation to changing transmission conditions, development of new technologies, communication security, and training and education. The study of modulation formats for telecommunication signals was carried out. A comparative analysis of the main modulation methods in wireless data transmission systems was performed. The optimization of the power spectral density of modulated telecommunication signals for the effective use of wireless communication systems is realized. The computer modeling of frequency shift keying (FSK) of radio signals in wireless communication systems is carried out. The aspects of ensuring safe and healthy working conditions for all employees are determined.

Keywords: modulation format; wireless transmission system; spectral power integrity; computer modeling.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	6
ВСТУП.....	8
1 ДОСЛІДЖЕННЯ ФОРМАТІВ МОДУЛЯЦІЇ ТЕЛЕКОМУНІКАЦІЙНИХ СИГНАЛІВ.....	11
1.1 Модуляція та демодуляція.....	11
1.2 Методи аналогової модуляції	13
1.3 Методи цифрової модуляції.....	15
1.4 Генерування тестових телекомунікаційних сигналів.....	21
1.5 Висновки до розділу 1	32
2 МЕТОДИ МОДУЛЯЦІЇ В БЕЗПРОВІДНИХ СИСТЕМАХ	34
2.1 Обґрунтування вибору формату модуляції телекомунікаційних сигналів...	34
2.2 Бінарна фазова маніпуляція (BPSK).....	37
2.3 Диференціальна модуляція BPSK (D-BPSK)	40
2.4 Методи безперервної фазової модуляції телекомунікаційних сигналів	43
2.5 Модуляція з безперервною фазовою маніпуляцією частоти (CPFSK).....	47
2.6 Технологія мінімальної частотної маніпуляції	51
2.7 Еквівалентність модуляції OQPSK.....	52
2.8 Дослідження властивостей фазових переходів.....	54
2.9 Висновки до розділу 2	64
3 СПЕКТРАЛЬНА ЩІЛЬНІСТЬ ПОТУЖНОСТІ МОДУЛЬОВАНИХ ТЕЛЕКОМУНІКАЦІЙНИХ СИГНАЛІВ.....	66
3.1 Методи оцінювання спектральної густини потужності сигналів	66
3.2 Лістинги програмного визначення спектральної щільності потужності (PSD).....	73
3.3 Висновки до розділу 3	78
4 КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ЧАСТОТНОЇ МАНІПУЛЯЦІЇ (FSK) РАДІОСИГНАЛІВ.....	79
4.1 Дискретна еквівалентна модель BFSK модуляції.....	79

	5
4.2 Когерентний демодулятор.....	82
4.3 Некогерентний демодулятор.....	84
4.4 Спектральна густина потужності	86
4.5 Висновки до розділу 4	88
5 ОХОРОНА ПРАЦІ	90
5.1 Технічні рішення щодо безпечного виконання роботи.....	90
5.1.1 Обладнання робочого місця.....	90
5.2 Технічні рішення з гігієни праці та виробничої санітарії.....	93
5.2.1 Мікроклімат	93
5.2.2 Склад повітря робочої зони.....	94
5.2.3 Виробниче освітлення.....	95
5.2.4 Виробничий шум.....	96
5.2.5 Виробничі випромінювання.....	96
5.3 Пожежна безпека.....	98
5.3.1 Технічні рішення системи запобігання пожежі	98
5.3.2 Технічні рішення системи протипожежного захисту.....	100
ВИСНОВКИ.....	101
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	103
ДОДАТКИ.....	106
Додаток А Ілюстративна частина.....	107
Додаток Б Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень	111

ПЕРЕЛІК СКОРОЧЕНЬ

3GPP - Проект партнерства третього покоління
6LoWPAN - IPv6 через WPAN з низькою потужністю
A- CSCF - Контролер границі сеансу доступу
ADSL - Асиметрична цифрова абонентська лінія
AM - Амплітудна модуляція
AMPS - Розширена система мобільного зв'язку
AMQP - протокол розширеної черги повідомлень
ANN - Штучна нейронна мережа
ASK - Амплітудна маніпуляція
ATM - Асинхронний режим передачі
AuC - Центр автентифікації
BBU - Блок базового діапазону
BER - Бітовий коефіцієнт помилок
BGP - Протокол граничного шлюзу
BICN - Базова мережа, що не залежить від носія
BSC - Контролер базової станції
BSS - Підсистема базової станції
BTS - Базова приймально-передавальна станція
CA - Агрегація несучих
CDM - мультиплексування з кодовим поділом каналів
CDMA - множинний доступ з кодовим поділом каналів
CDR - Детальний запис виклику
CINR - відношення несуча/завада + шум
CIR - Канальна імпульсна характеристика
CNN - Згорткова нейронна мережа
CoAP - протокол обмежених додатків
CPC - Безперервне пакетне з'єднання
C- RAN - хмарна/централізована мережа радіодоступу

CSCP - Функція управління сеансом зв'язку

DAS - розподілена антенна система

DC - подвійне підключення

DDoS - розподілена відмова в обслуговуванні

DDS - служба розподілу даних

DNS - система доменних імен

DPWS - Профіль пристроїв для веб-сервісів

D-RAN - Розподілена мережа радіодоступу

DSCP - Диференційована кодова точка обслуговування

DTLS - Безпека транспортного рівня дейтаграм

EDGE - Розширені швидкості передачі даних для GSM Evolution

EIR - Ідентифікаційний реєстр обладнання

eMBB - Розширений мобільний широкосмуговий зв'язок

EPC - вдосконалене пакетне ядро

EV-DO - Еволюція з оптимізацією даних

FDD - Дуплексування з частотним розділенням каналів

FDM - мультиплексування з частотним розділенням каналів

FDMA - множинний доступ з частотним розділенням каналів

FM - частотна модуляція

FSK - частотна маніпуляція

FTAM - Передача, доступ і керування файлами

FTP - протокол передачі файлів

GAN - Генеративна змагальна мережа

GEO - Геосинхронна екваторіальна орбіта

GERAN - гранична мережа радіодоступу GSM

GGSN - Шлюзовий вузол підтримки GPRS

GMSC - Центр мобільної комутації шлюзу

GPOS - Операційна система загального призначення

GPRS - Загальна служба пакетного радіозв'язку

GSM - Глобальна система мобільного зв'язку

ВСТУП

Актуальність теми. Дослідження модуляції в безпроводних системах за допомогою Python залишається дуже актуальним і важливим в наш час з наступних причин. Безпроводні технології постійно розвиваються, що вимагає постійного аналізу та вдосконалення методів модуляції для оптимізації ефективності та надійності передачі даних. Із зростанням популярності мобільних додатків, відео стрімінгу, Інтернету речей тощо, обсяги передаваних даних в безпроводних мережах значно збільшилися, що ставить під сумнів ефективність традиційних методів модуляції [1]. Зростаюча конкуренція за обмежений радіочастотний спектр підштовхує до розвитку нових методів модуляції, які дозволяють більш ефективно використовувати доступну спектральну ширину. Розвиток нових технологій, таких як 5G, Internet of Things (IoT), smart grids тощо, потребує адаптації методів модуляції для врахування їх унікальних вимог та обмежень. Споживачі очікують все більш швидкого та надійного з'єднання в умовах зростаючої кількості пристроїв та різноманітних типів сервісів, що вимагає подальшого вдосконалення методів модуляції.

Використання Python для дослідження модуляції в безпроводних системах надає дослідникам та інженерам потужний інструмент для аналізу, симуляції та оптимізації різних методів модуляції в реальному часі. Він дозволяє швидко реалізовувати та тестувати нові ідеї та концепції, що є критичним у швидкозмінному середовищі безпроводних комунікацій [2].

Модуляція та демодуляція є ключовими процесами у телекомунікаційних системах. Модуляція полягає в перетворенні інформації (наприклад, звукових сигналів або даних) на електричні сигнали, які можуть бути передані через канал зв'язку. Демодуляція, навпаки, це процес відновлення початкової інформації з отриманих електричних сигналів. Розуміння цих процесів дозволяє розкрити величезний потенціал телекомунікацій. Модуляція дозволяє ефективно передавати дані через різноманітні медіа (наприклад, радіохвилі, оптоволокну тощо), тоді як демодуляція допомагає отримати ці дані на приймачі. Це як

зрозуміти, як музика або мовлення перетворюються у радіохвилі, передаються через повітря, а потім знову перетворюються на звук на вашому радіоприймачі. Без цих процесів, передача і отримання інформації через великі відстані було б практично неможливим [3].

Аналіз останніх досліджень. Останні дослідження в області модуляції в безпроводних системах можуть бути важливими для розвитку нових технологій зв'язку. Python широко використовується для аналізу даних та моделювання в таких дослідженнях завдяки своїм багатofункціональним бібліотекам, таким як NumPy, SciPy, Matplotlib та іншим [1-3].

Основні кроки аналізу досліджень модуляції в безпроводних системах за допомогою Python. Зазвичай використовуються спеціальні прилади для запису сигналів, які потім аналізуються. Python може бути використаний для обробки цих даних. Використовуючи бібліотеки, такі як NumPy та SciPy, можна виконати різні операції, такі як фільтрація сигналу, розрахунок спектру сигналу тощо. Python також має багато інструментів для моделювання систем модуляції. Наприклад, бібліотека `'scikit-learn'` може бути використана для побудови класифікаторів, які визначають модуляцію сигналу. Моделі можуть бути протестовані на існуючих даних та оцінені їхні характеристики. За допомогою Matplotlib або інших бібліотек можна побудувати графіки, що демонструють характеристики сигналів або ефективність моделей. На основі аналізу даних і моделювання можна зробити висновки щодо ефективності різних методів модуляції в безпроводних системах та їхнього впливу на ефективність передачі даних. Наведений вище підхід може бути використаний для аналізу останніх досліджень в цій області з використанням Python [4-6].

Створення програмних застосунків для дослідження форматів модуляції в безпроводних системах передачі має кілька обґрунтувань. Різні модуляційні схеми мають різні характеристики, такі як швидкість передачі даних, стійкість до шуму, спектральна ефективність тощо. Програмні застосунки дозволяють проводити експерименти та аналізувати характеристики різних схем модуляції в різних умовах. Розуміння та аналіз форматів модуляції дозволяє оптимізувати

безпроводні системи передачі для досягнення кращої продуктивності, зниження споживання енергії, покращення якості зв'язку та інших показників. Вивчення форматів модуляції може стимулювати розробку нових технологій і підходів до безпроводної комунікації, що може мати значний вплив на розвиток сучасних мереж і пристроїв. Проведення досліджень та розробка програмних засобів для аналізу форматів модуляції може бути важливим компонентом у навчальних програмах з безпроводних комунікацій та телекомунікацій, де студенти можуть вивчати та вдосконалювати свої знання та вміння в цій галузі.

Мета та постановка задачі. Метою даної бакалаврської дипломної роботи є дослідження форматів модуляції в безпроводних системах передачі для підвищення ефективності передачі даних, адаптацію до змінних умов передачі, розвиток нових технологій, забезпечення безпеки зв'язку та навчання і освіти.

Задачами бакалаврської дипломної роботи є:

- дослідження форматів модуляції телекомунікаційних сигналів;
- порівняльний аналіз основних методів модуляції в безпроводних системах передачі даних;
- оптимізація спектральної щільності потужності модульованих телекомунікаційних сигналів для ефективного використання бездротових систем зв'язку;
- комп'ютерне моделювання частотної маніпуляції (FSK) радіосигналів в бездротових комунікаційних системах;
- визначення аспектів забезпечення безпечних та здорових умов праці для всіх працівників.

Апробація результатів роботи. Основні ідеї роботи доповідались і обговорювались на науковій конференції ВНТУ у 2024 році.

1 ДОСЛІДЖЕННЯ ФОРМАТІВ МОДУЛЯЦІЇ ТЕЛЕКОМУНІКАЦІЙНИХ СИГНАЛІВ

1.1 Модуляція та демодуляція

Перш ніж перейти до деталей техніки модуляції, спробуємо пояснити цю тему за допомогою аналогії. Уявіть, що у вас є аркуш паперу з інформацією. Модуляція - це процес перетворення інформації (наприклад, цифрових даних) на сигнал, який може бути переданий через канал зв'язку (наприклад, повітря або провід). У прикладі, загортання паперу у камінь відображає процес модуляції, де інформація кодується у фізичну форму, яка може бути передана.

Демодуляція, навпаки, це процес витягання інформації з сигналу, отриманого після передачі. У прикладі, коли ваш друг зішкребає інформацію з каменю, це є процесом демодуляції, де інформація відтворюється з фізичного носія. У реальних системах зв'язку модуляція відбувається шляхом зміни параметрів сигналу, таких як амплітуда, частота або фаза, для кодування інформації. Демодуляція зазвичай включає в себе процес виявлення і витягання цієї інформації з прийнятого сигналу.

Процес модуляції та демодуляції широко використовується в різних технологіях зв'язку, включаючи радіо, телебачення, мобільні телефони та Інтернет. Він дозволяє ефективно передавати дані на великі відстані і забезпечує їх правильне отримання та розуміння на приймаючому кінці (рис. 1.1) [7].

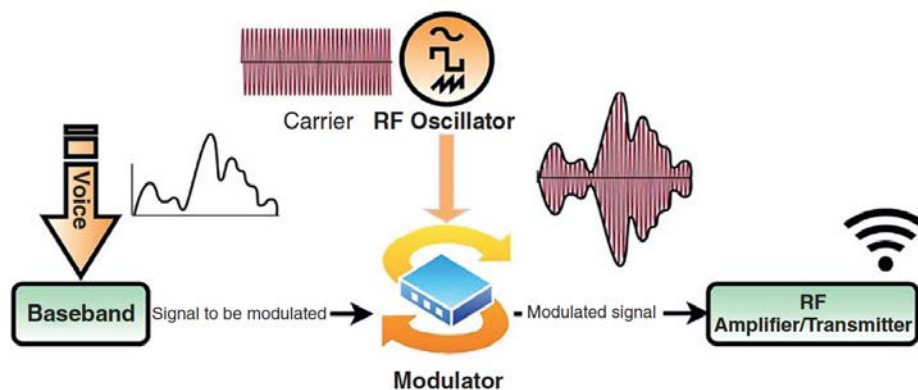


Рисунок 1.1 - Процес модулювання

Відштовхуючись від попереднього прикладу, ми можемо пояснити процес модуляції як надсилання інформаційного сигналу з меншою частотою і потужністю шляхом накладання інформаційного сигналу на сигнал-носії з набагато більшою інтенсивністю потужності і частотою. Електричні/електромагнітні сигнали в комунікаційному середовищі мають синусоїдальну форму. Ми можемо говорити про три змінні величини цього синусоїдального сигналу: амплітуду, частоту і фазу. Ми здійснюємо процес модуляції, коли змінюємо амплітуду, частоту або фазу цього синусоїдального сигналу в залежності від амплітуди інформаційного сигналу, який ми будемо передавати. Якщо сигнал надходить з протилежного боку, інформаційний сигнал отримується шляхом зворотного процесу.

Цей процес називається демодуляцією. Процеси модуляції та демодуляції виконуються пристроями, які називаються модемами (модулятор-емодулятор) (рис. 1.2) [8].

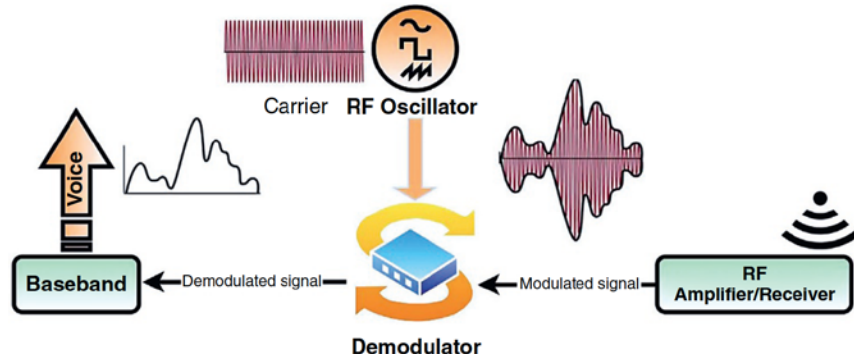


Рисунок 1.2 - Процес демодуляції

У наступних розділах буде описано три основні методи модуляції з використанням цих параметрів (тобто амплітуди, частоти і фази).

1.2 Методи аналогової модуляції

В аналогових методах модуляції інформаційний сигнал у синусоїдальній формі передається шляхом «накладання» на зміну амплітуди, частоти або фази несучої, яка також має синусоїдальну форму. Система визначається як аналогова, оскільки інформаційний сигнал має аналогову (синусоїдальну) форму.

В амплітудній модуляції амплітуда несучого сигналу є функцією інформаційного сигналу. Іншими словами, амплітуда несучого сигналу формується формою інформаційного сигналу. Амплітуда несучого сигналу змінюється залежно від зміни інформаційного сигналу. Ці зміни амплітуди виявляються в модемі, розташованому на протилежному кінці. Інформаційний сигнал знову отримується у своїй початковій формі (рис. 1.3) [9].

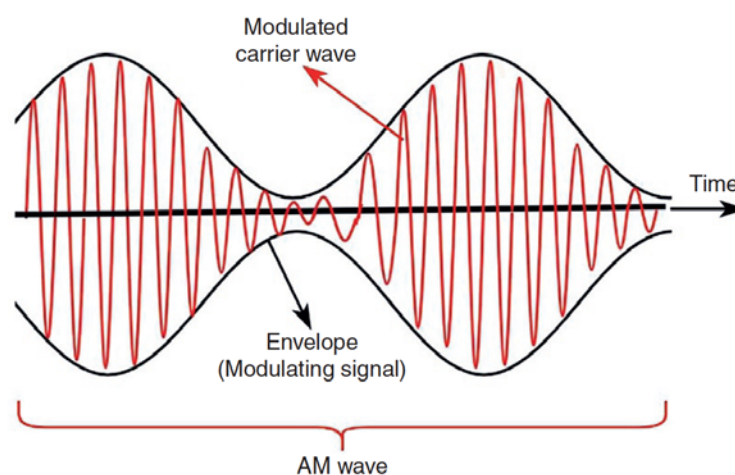


Рисунок 1.3 - Амплітудна модуляція

При частотній модуляції частота несучого сигналу є функцією інформаційного сигналу. Іншими словами, частота несучого сигналу формується формою інформаційного сигналу. Частота несучого сигналу змінюється в залежності від зміни інформаційного сигналу. Інформаційний сигнал відновлюється в початковій формі шляхом виявлення цих змін частоти в модемі на протилежному кінці телекомунікаційного тракту передавання (рис. 1.4) [10].

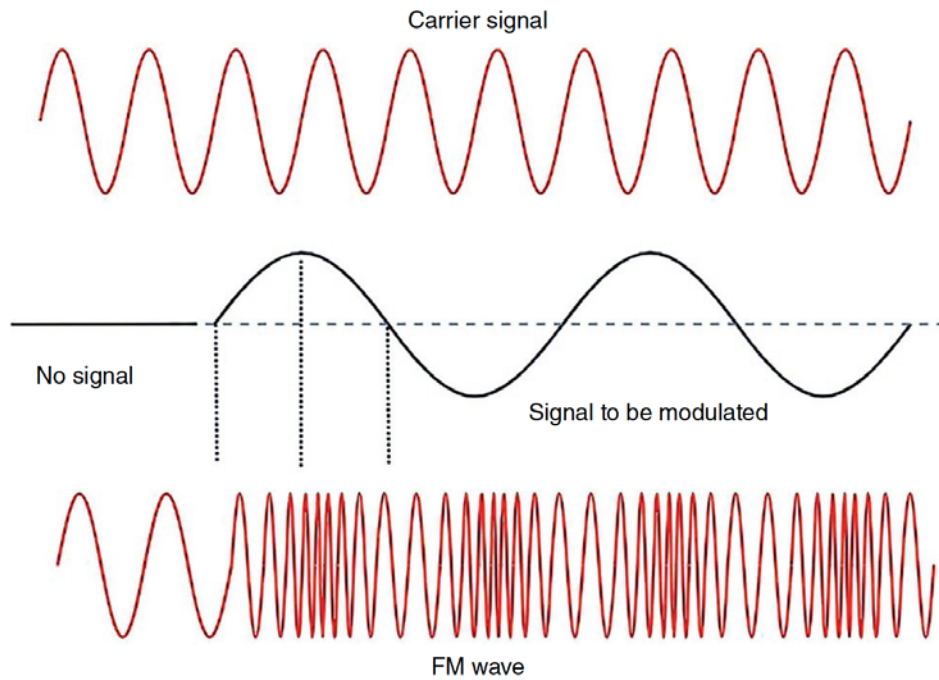


Рисунок 1.4 - Частотна модуляція (ЧМ)

У фазовій модуляції фаза несучого сигналу є функцією інформаційного сигналу. Іншими словами, фаза несучого сигналу формується відповідно до форми інформаційного сигналу. Фаза несучого сигналу змінюється залежно від зсуву інформаційного сигналу. Ці зміни фази виявляються в модемі на протилежному кінці, і інформаційний сигнал відновлюється в його первісній формі (рис. 1.5) [11].

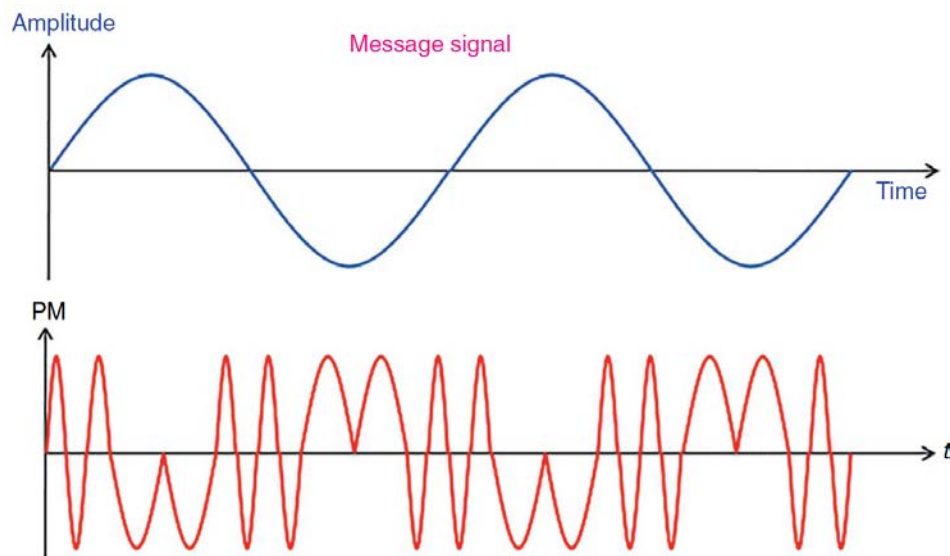


Рисунок 1.5 - Фазова модуляція

1.3 Методи цифрової модуляції

В аналогових системах модуляції інформаційний сигнал має аналогову форму. З іншого боку, в системах цифрової модуляції інформаційний сигнал має форму переходів або квадратичних змін амплітуди, що виражають 1s і 0s. Ця цифрова форма інформації отримується (особливо при передачі аудіо та відео) шляхом застосування аналого-цифрового перетворення інформаційного сигналу, який спочатку був створений як аналоговий. Тобто, аналоговий інформаційний сигнал перетворюється у форми 1 і 0 шляхом цифрового перетворення. На протилежному кінці, вихідний аналоговий інформаційний сигнал отримується шляхом цифро-аналогового перетворення.

Як показано на рисунку 1.6, в точках, де амплітуда інформаційного сигналу позитивна, є амплітуда несучого сигналу. З іншого боку, при від'ємних значеннях амплітуди інформаційного сигналу амплітуда несучого сигналу зменшується до 0 рівнів (100% модуляція). При 50% модуляції 50% амплітуди несучого сигналу приймається в точках, де інформаційний сигнал дорівнює нулю (рис. 2.6) [12].

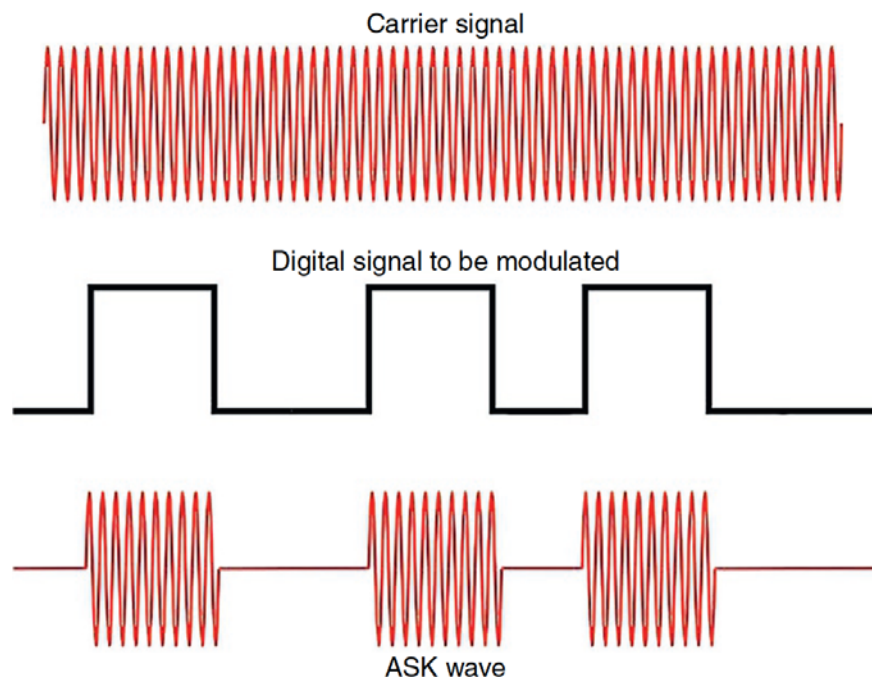


Рисунок 1.6 - Амплітудна маніпуляція (ASK)

Інформаційний сигнал має два відмінних значення амплітуди - позитивне і негативне. У цьому випадку несучий сигнал несе інформацію, вводячи дві різні частотні форми. Для значень 0 використовується частота x , для 1 - частота y ($x < y$) (рис. 1.7) [1].

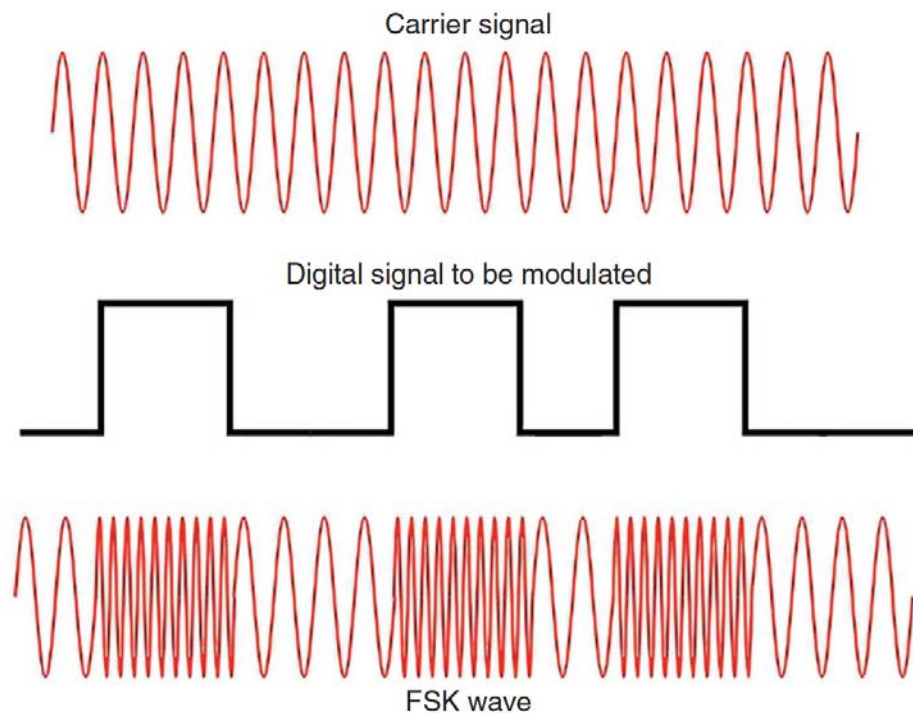


Рисунок 1.7 - Частотна маніпуляція (FSK)

У методі фазової маніпуляції фаза несучого сигналу також змінюється щоразу, коли амплітуда інформаційного сигналу змінюється з додатної на від'ємну або з від'ємної на додатну. Як згадувалося в попередніх розділах, ці фазові зміни виявляються на протилежному кінці, і отримується вихідний інформаційний сигнал (рис. 1.8) [2].

Розуміння квадратурної амплітудної модуляції (QAM), яка стала стандартом у мережах 4G та інших бездротових мережах зв'язку, має вирішальне значення для розуміння цих систем. Тому ми пояснимо цю схему модуляції більш детально.

У попередніх розділах ми говорили про фазову та амплітудну модуляцію. QAM можна визначити як комбінацію цих двох типів модуляції. Іншими

словами, в техніці QAM інформація передається як в амплітуді, так і в фазі несучого сигналу. Таким чином, ми отримуємо вдвічі більшу пропускну здатність.

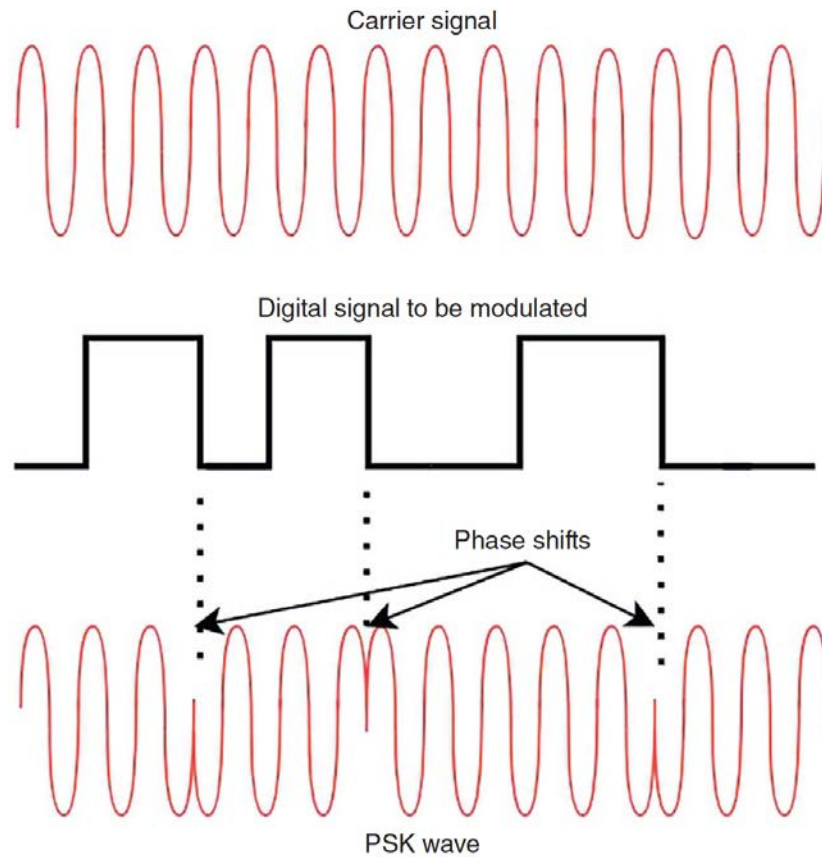


Рисунок 1.8 - Фазова маніпуляція (PSK)

Виходячи з цього визначення, QAM відоме як квадратурне мультиплексування з несучою [3].

У QAM використовуються два несучі сигнали однакової частоти, але з різницею фаз 90° у найпростішій формі. Сигнал, що починається з фази 0° , називається «I», а інший сигнал у фазі 90° називається «Q».

Враховуючи, що між синусоїдальним і косинусоїдальним сигналами існує різниця фаз 90° , ми можемо одночасно передавати обидва несучі сигнали на одній частоті з середовища передачі (рис. 1.9). Як базова концепція, метод,

показаний на рисунку 1.9, є аналоговим QAM і використовується для передавання кольорової інформації в аналоговому телевізійному мовленні [4].

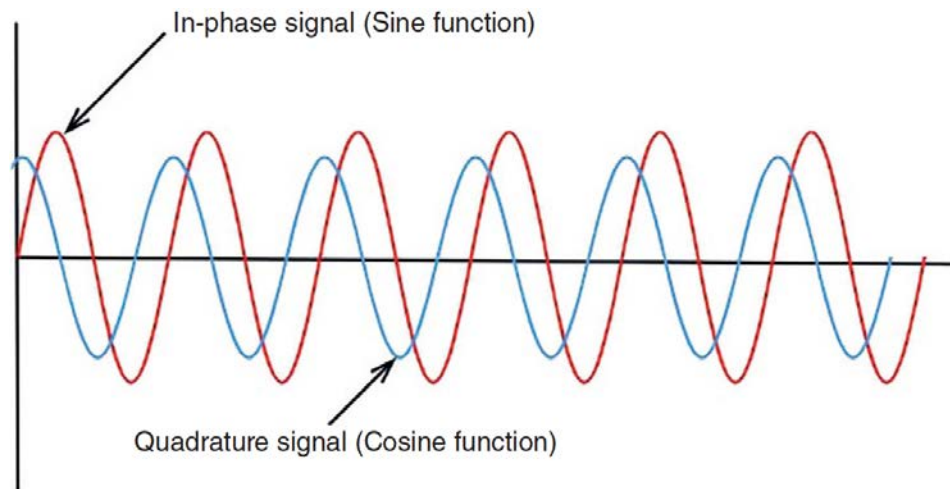


Рисунок 1.9 - Синфазний сигнал і квадратурна складова сигналу

Квадратурна амплітудна модуляція (QAM) відома своєю ефективністю та широким застосуванням у стільниковому зв'язку та бездротових технологіях, зокрема в Wi-Fi.

У порівнянні з іншими методами модуляції, такими як амплітудна та фазова модуляція, цифрова QAM надає додаткову гнучкість та ефективність. Це досягається шляхом одночасної зміни амплітуди і фази сигналу, що дозволяє передавати більше даних у кожному символі. Оскільки QAM здатна передавати більше інформації в один символ, це дозволяє досягати високої швидкості передачі даних. Наприклад, в Wi-Fi QAM використовується для досягнення високошвидкісного бездротового зв'язку у стандартах 802.11n, 802.11ac та 802.11ax. Крім високошвидкісної передачі даних, QAM також відома своєю здатністю до самокорекції та завадостійкості. За допомогою спеціальних алгоритмів корекції помилок і технік обробки сигналу QAM може ефективно виявляти та виправляти помилки, що дозволяє підтримувати стабільний зв'язок навіть у шумних середовищах. Отже, цифрова QAM виявляється надзвичайно

корисною у бездротовому зв'язку та забезпечує високу швидкість передачі даних, низькі рівні помилок та високу стійкість до завад.

У цифрових схемах квадратурної амплітудної модуляції (QAM) кількість бітів, які можуть бути передані одночасно, може бути збільшена за допомогою різних значень фази та амплітуди. Це може бути представлено у вигляді діаграми сузір'я, де кожна точка на сузір'ї відповідає певному комбінації значень фази та амплітуди.

У найпростішій формі QAM може бути використана для передачі одного біту інформації одночасно. Це відомо як QPSK (Quadrature Phase Shift Keying), де фаза сигналу може бути відмінною лише на 90 градусів, що дозволяє передавати два біти (00, 01, 10, 11) на кожній символній період. За допомогою більш складних схем QAM можна передавати більше бітів одночасно. Наприклад, в 16-QAM кожній точці сузір'я відповідає 4 бітам (0000, 0001, ..., 1111), оскільки є 16 різних комбінацій амплітуди та фази. Отже, використання QAM дозволяє ефективно використовувати доступну пропускну здатність каналу для передачі більшої кількості інформації одночасно, що робить його ключовою технологією для високошвидкісних цифрових комунікаційних систем [5].

Використання багаторівневих схем QAM (наприклад, 16-QAM, 32-QAM, 64-QAM тощо) дозволяє передавати більше бітів інформації за один символний період, забезпечуючи високу пропускну здатність та ефективне використання каналу передачі даних. У схемі 16-QAM використовується 16 різних точок у сузір'ї, що відповідають 16 різним комбінаціям амплітуди та фази. Це дозволяє передавати 4 біти інформації одночасно [6].

Також інші багаторівневі схеми QAM, такі як 32-QAM, 64-QAM, 128-QAM та 256-QAM, дозволяють передавати ще більше інформації за символний період. Наприклад, в схемі 64-QAM кожному символу може відповідати 6 біт інформації, що дозволяє передавати 64 різних символи (рис. 1.10). Проте, слід зазначити, що зі збільшенням кількості рівнів QAM, сигнал стає більш чутливим до шуму та спотворень, що може призвести до погіршення якості прийому. Тому,

при виборі рівня модуляції потрібно збалансувати між пропускнуою здатністю та стійкістю до помилок [7].

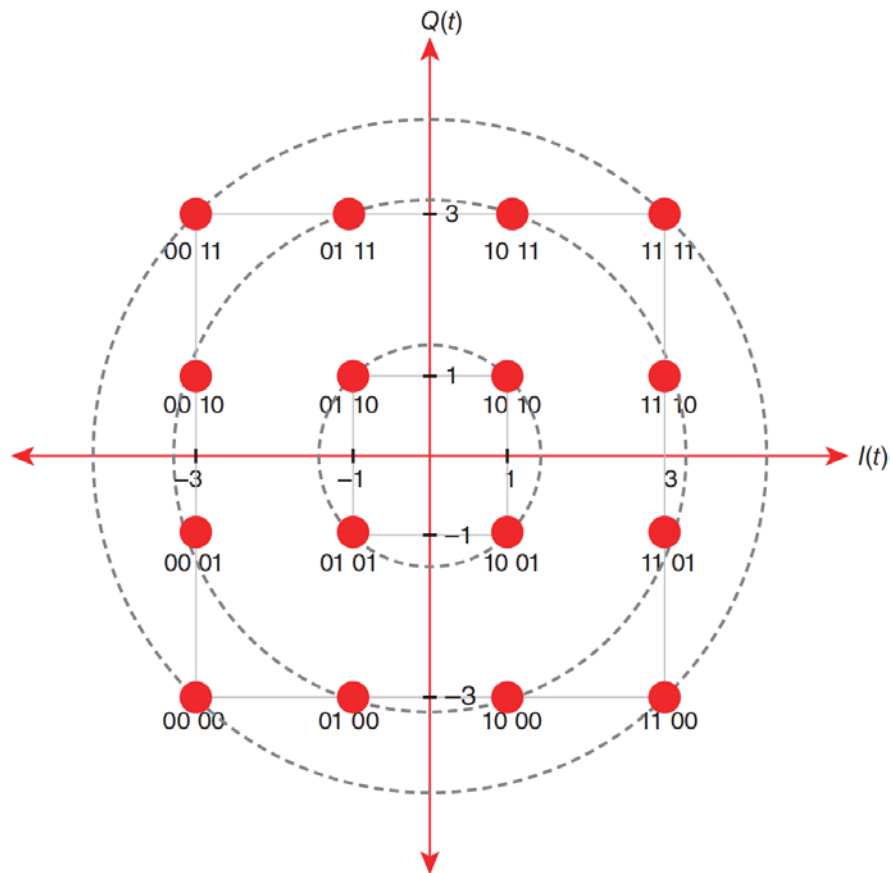


Рисунок 1.10 - Діаграма сузір'я QAM

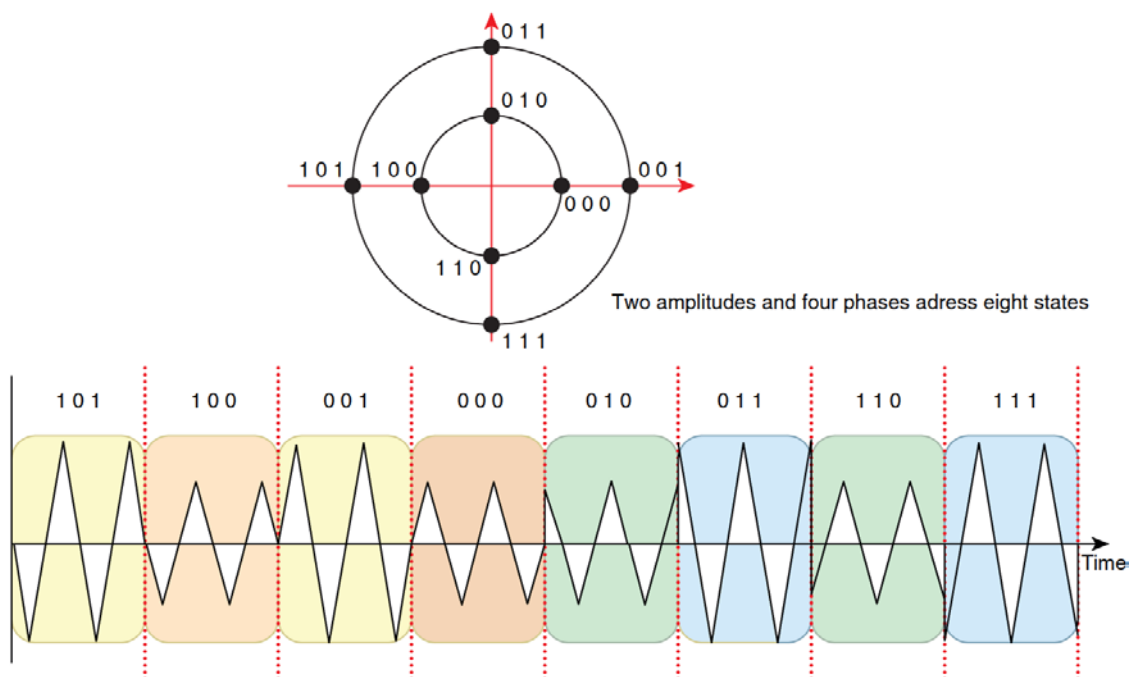


Рисунок 1.11 - 8-QAM сигнал

Можна сказати, що в діаграмі 16-QAM є 3 різні амплітуди (евклідова відстань від кожної точки до початку координат) і 12 різних фаз (кут, утворений кожним кінцем). Іншими словами, дивлячись на діаграму, ми можемо сказати, що відстані кожної полярної координати до початку координат мають 1 з трьох різних значень.

Дивлячись на діаграму знову, ми можемо сказати, що кожна полярна координата знаходиться в 1 з 12 фаз (кутів). Наприклад, можна сказати, що символ «1111» буде передано з несучою з амплітудою 2 і фазою 45° . Приклад зображення 8-QAM сигналу наведено на рисунку 1.11 [8].

1.4 Генерування тестових телекомунікаційних сигналів

Використання простих тестових сигналів, таких як синусоїдальні, прямокутні імпульси, гаусівські імпульси та чірпування, є стандартною практикою в експериментальному моделюванні та імітації систем зв'язку. Ці сигнали можуть бути використані для тестування та верифікації алгоритмів обробки сигналів, методів демодуляції, а також для оцінки різних параметрів та характеристик системи.

Існує кілька способів генерації цих тестових сигналів в Python. Один із способів - написати власні функції для генерації сигналів, що відповідають вищезазначеним типам. Проте, щоб уникнути зайвого кодування, можна скористатися бібліотеками Python, такими як SciPy, які вже містять функції для генерації різних типів сигналів.

Наприклад, бібліотека SciPy містить функції для генерації синусоїдальних сигналів (`scipy.signal.square`), прямокутних імпульсів (`scipy.signal.square`), гаусівських імпульсів (`scipy.signal.gausspulse`) та чірпувань (`scipy.signal.chirp`). Використання цих функцій дозволяє легко створювати різні тестові сигнали для експериментального моделювання та імітації систем зв'язку в Python.

Наприклад, для створення синусоїдального сигналу можна скористатися наступним кодом:

```

import numpy as np
import matplotlib.pyplot as plt

# Параметри сигналу
amplitude = 1
frequency = 10 # Гц
duration = 1 # секунди
sampling_rate = 1000 # частота дискретизації (кількість вимірів на секунду)

# Генерація часового вектора
t = np.linspace(0, duration, int(sampling_rate * duration), endpoint=False)

# Генерація синусоїдального сигналу
signal = amplitude * np.sin(2 * np.pi * frequency * t)

# Відображення сигналу
plt.plot(t, signal)
plt.xlabel('Час (с)')
plt.ylabel('Амплітуда')
plt.title('Синусоїдальний сигнал')
plt.show()

```

Аналогічно, можна скористатися функціями ``scipy.signal.square``, ``scipy.signal.gausspulse`` та ``scipy.signal.chirp`` для генерації прямокутних імпульсів, гаусівських імпульсів та чірпувань відповідно [9].

Отже, для того, щоб згенерувати синусоїду, першим кроком є встановлення частоти f синусоїди. Наприклад, ми хочемо згенерувати синусоїду з частотою $f = 10$ Гц, мінімальна та максимальна амплітуди якої становлять -1В та $+1\text{В}$ відповідно.

Знаючи частоту синусоїди, наступним кроком є визначення частоти дискретизації.

Для сигналів основної смуги частота дискретизації є прямою. Згідно з теоремою дискретизації Найквіста-Шеннона, для точного відтворення неперервного сигналу в дискретній області, необхідно дискретизувати сигнал з частотою f_s , що принаймні вдвічі перевищує максимальну частоту f_m , яка міститься в сигналі (фактично, це вдвічі більша одностороння смуга пропускання, яку займає реальний сигнал). Для базового сигналу смуга пропускання (від 0 до f_m) і максимальна частота f_m в даній смузі еквівалентні) [10].

Напишемо функцію для генерування синусоїдального сигналу за допомогою бібліотеки Numpy мови Python. Numpy - це фундаментальна бібліотека для наукових обчислень у мові Python. Для того, щоб використовувати пакет numpy, його потрібно імпортувати. У цьому прикладі ми імпортуємо пакунок numpy та перейменовуємо його на коротший псевдонім np.

```
import numpy as np
```

Далі визначаємо функцію для генерації синусоїдального сигналу з необхідними параметрами.

```
def sine_wave(f, overSampRate, phase, nCyl):
    """
    Generate sine wave signal with the following parameters
    Parameters:
        f : frequency of sine wave in Hertz
        overSampRate : oversampling rate (integer)
        phase : desired phase shift in radians
        nCyl : number of cycles of sine wave to generate
    Returns:
        (t,g) : time base (t) and the signal g(t) as tuple
    Example:
        f=10; overSampRate=30;
        phase = 1/3*np.pi; nCyl = 5;
        (t,g) = sine_wave(f, overSampRate, phase, nCyl)
    """
    fs = overSampRate*f # sampling frequency
    t = np.arange(0, nCyl*1/f-1/fs, 1/fs) # time base
    g = np.sin(2*np.pi*f*t+phase) # replace with cos if a cosine wave is desired
    return (t,g) # return time base and signal g(t) as tuple
```

Зауважимо, що функцію sine wave визначено у файлі з назвою signalgen.py. Ми додамо більше подібних функцій у той самий файл. Мета полягає у тому, щоб тримати всі пов'язані функції генерування сигналів в одному файлі. Цей підхід може бути поширено на об'єктно-орієнтоване програмування. Тепер, коли ми визначили синусоїдальну функцію у signalgen.py, все, що нам потрібно зробити, це викликати її з необхідними параметрами і побудувати графік [11].

```
def sine_wave_demo():
    """
    Simulate a sinusoidal signal with given sampling rate
    """
    import numpy as np
    import matplotlib.pyplot as plt # library for plotting
    from DigiCommPy.signalgen import sine_wave # import the function

    f = 10 #frequency = 10 Hz
    overSampRate = 30 #oversampling rate
    phase = 1/3*np.pi #phase shift in radians
    nCyl = 5 # desired number of cycles of the sine wave
    (t,g) = sine_wave(f,overSampRate,phase,nCyl) #function call

    plt.plot(t,g) # plot using pyplot library from matplotlib package
    plt.title('Sine wave f='+str(f)+' Hz') # plot title
    plt.xlabel('Time (s)') # x-axis label
    plt.ylabel('Amplitude') # y-axis label
    plt.show() # display the figure
```

Python - це мова програмування на основі інтерпретатора, яка обробляє все в цифровому вигляді. Для того, щоб отримати гладку синусоїду, частота дискретизації повинна бути набагато вищою за мінімально необхідну, тобто принаймні вдвічі перевищувати частоту f - згідно з теоремою Найквіста-Шеннона. Отже, нам потрібно дискретизувати вхідний сигнал зі швидкістю, значно вищою за ту, що диктує критерій Найквіста. Вища частота передискретизації вимагає більшого обсягу пам'яті для зберігання сигналу. Бажано, щоб коефіцієнт передискретизації не перевищував прийнятного значення.

У попередній функції вибрано коефіцієнт передискретизації 30. Це робиться для того, щоб побудувати плавну безперервну синусоїду. Таким чином, частота дискретизації стає $f_s = 30 \times f = 30 \times 10 = 300$ Гц. Якщо для синусоїди потрібен зсув фази, вкажіть його також [12].

У Python функцію можна швидко завантажити і виконати за допомогою наступних виразів у командному рядку.

```
>> import sys
>> sys.path.append('path_where_digicompy_resides') #search path
>> from DigiCommPy.chapter_1.demo_scripts import sine_wave_demo #import function
>> sine_wave_demo() #call the function to execute it
```

Результуючий графік наведено на рисунку 1.12.

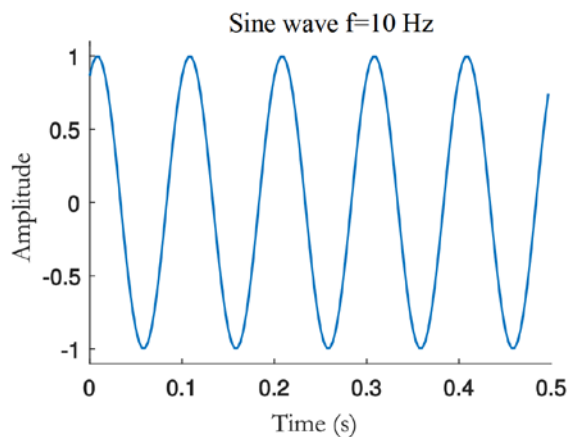


Рисунок 1.12 - Синусоїдальний сигнал 10 Гц з 5 циклами та фазовим зсувом $1/3\pi$ радіан

Найбільш логічним способом передачі інформації через канал зв'язку є потік прямокутних імпульсів - окремий імпульс для «0» та інший для «1». Цифрові сигнали графічно зображуються у вигляді прямокутних хвиль з певним періодом між символами/бітами. Синусоїдальні сигнали також широко використовуються в комутаційних схемах, як тактові сигнали, що синхронізують різні блоки

цифрових схем, як еталонний тактовий генератор для певної області системи тощо [6].

Прямокутний сигнал проявляється у вигляді широкого спектру гармонік у частотній області і тому може спричиняти електромагнітні завади. Прямокутні сигнали є періодичними і містять непарні гармоніки при розкладанні в ряд Фур'є (тоді як такі сигнали, як пилкоподібні та інші сигнали дійсного слова, містять гармоніки на всіх цілих частотах). Оскільки прямокутний сигнал буквально розкладається на нескінченну кількість непарних гармонік у частотній області, апроксимація прямокутної послідовності є ще однією сферою інтересу. Кількість членів розкладу в ряд Фур'є, що використовується для апроксимації прямокутного сигналу, часто розглядається як явище Гіббса, яке проявляється у вигляді ефекту дзвону на кутах прямокутної хвилі в часовій області.

Справжні прямокутні сигнали - це особливий клас прямокутних послідовностей з 50% шпаруватістю. Зміна робочого циклу прямокутної хвилі призводить до широтно-імпульсної модуляції, де інформація передається шляхом зміни робочого циклу кожного прямокутного сигналу, що передається. Справжній прямокутний сигнал можна просто згенерувати, наклавши функцію `signum numpy.sign` на періодичну функцію.

```
def square_wave(f, overSampRate, nCyl):
    """
    Generate square wave signal with the following parameters
    Parameters:
        f : frequency of square wave in Hertz
        overSampRate : oversampling rate (integer)
        nCyl : number of cycles of square wave to generate
    Returns:
        (t,g) : time base (t) and the signal g(t) as tuple
    Example:
        f=10; overSampRate=30;nCyl = 5;
        (t,g) = square_wave(f,overSampRate,nCyl)
    """
    fs = overSampRate*f # sampling frequency
    t = np.arange(0,nCyl*1/f-1/fs,1/fs) # time base
    g = np.sign(np.sin(2*np.pi*f*t)) # replace with cos if a cosine wave is desired
    ↵
    return (t,g) # return time base and signal g(t) as tuple
```

Також можливо використати вбудовану функцію `square` з бібліотеки `SciPy` для генерування періодичного квадратичного сигналу із заданою шпаруватістю (рис. 1.13). Приклад коду, показаний нижче, генерує прямокутний сигнал $f = 10$ Гц з 20% шпаруватістю.

```
def scipy_square_wave():
    """
    Generate a square wave with given sampling rate
    """
    import numpy as np
    import matplotlib.pyplot as plt
    from scipy import signal

    f = 10 # f = 10Hz

    overSampRate = 30 # oversampling rate
    nCyl = 5 # number of cycles to generate

    fs = overSampRate*f # sampling frequency
    t = np.arange(start=0, stop=nCyl*1/f, step=1/fs) # time base
    g = signal.square(2 * np.pi * f * t, duty = 0.2)
    plt.plot(t,g); plt.show()
```

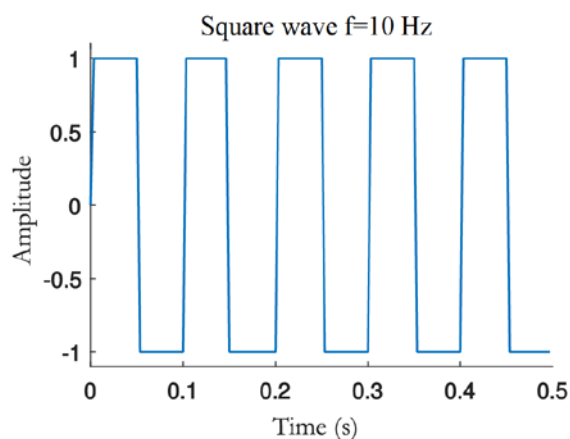


Рисунок 1.13 - Прямокутний сигнал 10 Гц з 5 циклами і 50% робочим циклом

Наступна функція імітує ізольований прямокутний імпульс амплітудою A із заданою тривалістю T , а результуючий графік показано на рисунку 1.14.

```
def rect_pulse(A, fs, T):
    """
    Generate isolated rectangular pulse with the following parameters
```

Parameters:

A : amplitude of the rectangular pulse
 fs : sampling frequency in Hz
 T : duration of the pulse in seconds

Returns:

(t,g) : time base (t) and the signal g(t) as tuple

Example:

```
A = 1; fs=500;T = 0.2;
(t,g) = rect_pulse(A,fs,T)
"""
t = np.arange(-0.5,0.5,1/fs) # time base
rect = (t > -T/2) * (t < T/2) + 0.5*(t == T/2) + 0.5*(t == -T/2)
g = A*rect
return (t,g)
```

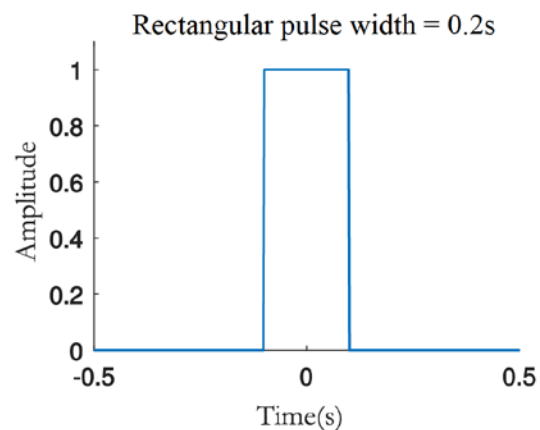


Рисунок 1.14 - Прямокутний імпульс з тривалістю 0,2 с

Гаусові фільтри грають важливу роль у цифровому зв'язку, особливо у системах з мінімальною маніпуляцією, таких як гаусова мінімальна маніпуляція (GMSK) та гаусова частотна маніпуляція (GFSK). Головна перевага використання гаусових фільтрів у цих системах полягає в їх властивостях м'якого переходу між станами сигналу, що дозволяє знизити спектральні бокові смуги та мінімізувати міжсимвольну інтерференцію.

У сфері обробки зображень гаусові фільтри також використовуються для створення ефекту гаусівського розмиву, що дозволяє зменшити шум та покращити різкість зображення. Вони дозволяють створювати плавні переходи між пікселями, що сприяє покращенню якості зображення [7].

У цілому, гаусові фільтри є важливими інструментами для обробки сигналів у цифровому зв'язку та обробки зображень, забезпечуючи ефективну передачу даних та покращення якості обробки зображень.

Наступна функція генерує гаусівський імпульс з $\sigma = 0,1$ с.

```
def gaussian_pulse(fs,sigma):
    """
    Generate isolated Gaussian pulse with the following parameters
    Parameters:
        fs : sampling frequency in Hz
        sigma : pulse width in seconds
    Returns:
        (t,g) : time base (t) and the signal g(t) as tuple
    Example:
        fs = 80; sigma = 0.1;
        (t,g) = gaussian_pulse(fs,sigma)
    """
    t = np.arange(-0.5,0.5,1/fs) # time base
    g = 1/(np.sqrt(2*np.pi)*sigma)*(np.exp(-t**2/(2*sigma**2)))
    return(t,g) # return time base and signal g(t) as tuple
```

Результуючий графік показано на рисунку 1.15.

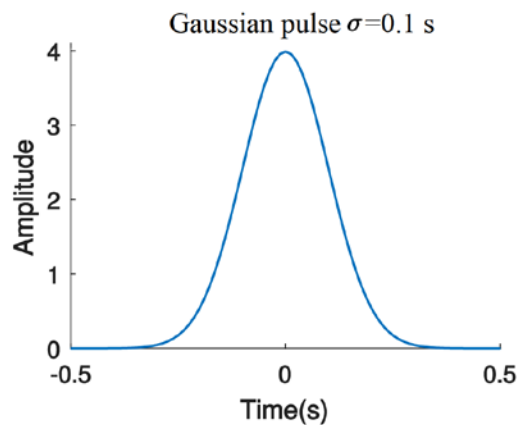


Рисунок 1.15 - Гаусівський імпульс з $\sigma = 0,1$ с

Всі сигнали, про які ми говорили до цього часу, не змінюють свою частоту з часом. Отримання сигналу зі змінною в часі частотою є основним завданням при розробленні моделі телекомунікаційного обладнання мобільної мережі.

Отже, чірп-сигнали є важливими у телекомунікаційних системах, особливо у радіолокаційних системах та системах передачі даних. Чірп-сигнал - це сигнал, частота якого змінюється з часом. Чірпи можуть мати різні форми, такі як лінійні, квадратичні або нелінійні, і можуть змінюватися від низьких до високих частот (up-chirp) або від високих до низьких частот (down-chirp).

Основні характеристики чірп-сигналів включають наступне.

Зміна частоти з часом є основною властивістю чірп-сигналів, яка дозволяє їм забезпечувати велику стійкість до шуму та інтерференції, а також високу роздільну здатність при радіолокаційних застосуваннях. Чірп-сигнали зазвичай мають широкосмуговий спектр, що дозволяє їм передавати багато інформації за короткий проміжок часу. Чірп-сигнали можуть бути менш чутливими до канальних ефектів, таких як затухання та розсіювання, через їх широкосмуговий характер. Чірп-сигнали використовуються у радіолокаційних системах для визначення відстані до об'єктів, а також у вимірювальних приладах для визначення характеристик об'єктів та середовища [9].

Створення та обробка чірп-сигналів вимагає спеціалізованих алгоритмів та обладнання, таких як чірп-радары та цифрові сигнальні процесори, що дозволяють генерувати, передавати та обробляти ці сигнали для різних застосувань у телекомунікаційних системах.

Лінійний чірп-сигнал лінійно змінює частоту від низьких до високих частот (або навпаки). Один із способів генерування чірп-сигналу полягає в об'єднанні серії сегментів синусоїдальних хвиль, кожна з яких має зростаючу (або спадаючу) частоту в певному порядку. Цей метод вносить розриви в чірп-сигнал через невідповідність фаз кожного з таких сегментів. Кращим підходом є модифікація рівняння синусоїди для генерування чірп-сигналу. Одним з методів є використання функції `chirp` з бібліотеки `SciPy` в `Python`. Використання функції `chirp` дозволяє згенерувати лінійний чірп-сигнал з вказаною початковою та кінцевою частотами, а також тривалістю сигналу. Наприклад, для створення лінійного чірп-сигналу від 100 Гц до 1000 Гц з тривалістю 1 секунди, можна скористатися наступним кодом:

```

import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import chirp

# Параметри чірп-сигналу
f0 = 100 # Початкова частота (в герцах)
f1 = 1000 # Кінцева частота (в герцах)
t = np.linspace(0, 1, 1000) # Час (в секундах)

# Генерація лінійного чірп-сигналу
signal = chirp(t, f0=f0, f1=f1, t1=1, method='linear')

# Візуалізація сигналу
plt.figure(figsize=(10, 5))
plt.plot(t, signal)
plt.title('Лінійний чірп-сигнал')
plt.xlabel('Час (с)')
plt.ylabel('Амплітуда')
plt.grid(True)
plt.show()

```

У цьому коді функція `chirp` генерує лінійний чірп-сигнал за вказаними параметрами: f_0 - початкова частота, f_1 - кінцева частота, t_1 - тривалість сигналу. Аргумент `method='linear'` вказує, що зміна частоти відбувається лінійно з часом.

Наведений лістинг код відображає створений лінійний чірп-сигнал за допомогою бібліотеки `Matplotlib`.

Отже, сигнал чірпування можна легко згенерувати за допомогою функції чірпування `Scipy`. Наступний скрипт використовує бібліотеку `Scipy` і генерує чірп з початковою частотою $f_0 = 1$ Гц на початку часової бази і $f_1 = 20$ Гц при $t_1 = 0,5$ с.

```
def chirp_demo():
    """
    Generating and plotting a chirp signal
    """
    import numpy as np
    import matplotlib.pyplot as plt
    from scipy.signal import chirp

    fs = 500 # sampling frequency in Hz
    t = np.arange(start = 0, stop = 1, step = 1/fs) #total time base from 0 to 1 second
    g = chirp(t, f0=1, t1=0.5, f1=20, phi=0, method='linear')
    plt.plot(t,g); plt.show()
```

Результуючий графік показано на рисунку 1.16.

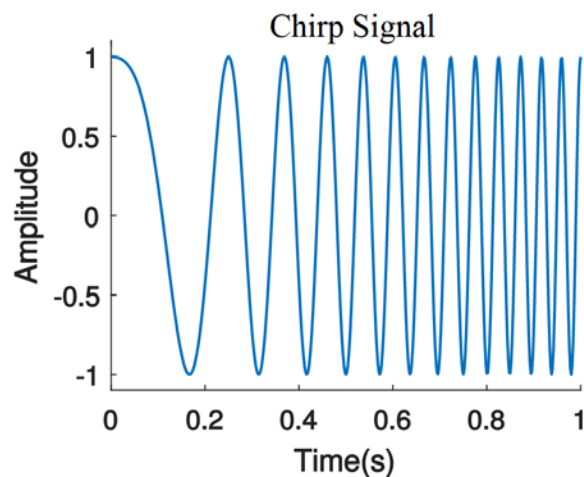


Рисунок 1.16 – Часова діаграма чірп-сигналу

1.5 Висновки до розділу 1

Результати дослідження форматів модуляції телекомунікаційних систем з використанням Python можуть бути дуже цінними для розуміння та оптимізації процесу передачі даних у безпроводних комунікаціях. Використовуючи Python, можна оцінити швидкість передачі даних для різних форматів модуляції та різних умов передачі. Результати можуть бути візуалізовані у вигляді графіків або діаграм для зручного порівняння різних форматів. Python дозволяє створювати моделі каналів передачі з різними параметрами, такими як шум,

спотворення, затухання сигналу тощо. Це допомагає вивчити, як різні формати модуляції впливають на стійкість передачі даних в різних умовах каналу передачі. Python може бути використаний для аналізу спектральної ефективності різних форматів модуляції. Це важливо для визначення оптимального використання доступної спектральної ширини та мінімізації міжканальних перешкод. Python може бути використаний для розрахунку параметрів якості передачі, таких як BER (бітова помилка) та SNR (відношення сигнал/шум). Це допомагає визначити, які формати модуляції є найбільш ефективними з точки зору надійності передачі.

Використовуючи Python, можна провести експерименти з різними параметрами системи, такими як розмір символу, ширина смуги, рівні модуляції тощо, для визначення оптимальних значень цих параметрів. Результати дослідження з допомогою Python можуть бути корисними для розробки та вдосконалення бездротових комунікаційних систем, а також для прийняття обґрунтованих рішень у телекомунікаційній галузі.

2 МЕТОДИ МОДУЛЯЦІЇ В БЕЗПРОВІДНИХ СИСТЕМАХ

2.1 Обґрунтування вибору формату модуляції телекомунікаційних сигналів

Моделювання смуги пропускання для різних методів модуляції важливе для оцінки їх продуктивності та вибору найбільш ефективного методу для конкретних застосувань. Наведемо короткий опис кожного методу модуляції та їх характеристики щодо спектральної ефективності та продуктивності.

BPSK використовує два можливих фазових стани для кодування даних, які відповідають двом значенням біта. Вона має просту реалізацію та низьку складність, але вимагає великої енергії на один біт. Диференційно кодована BPSK використовує різницю між сусідніми символами для кодування даних. Він є більш стійким до змін фази сигналу та має просту реалізацію. QPSK подібна до BPSK, але використовує чотири фазових стани для кодування двох бітів одночасно. Вона має вищу спектральну ефективність, але більшу складність. Зміщена QPSK дозволяє уникнути появи нулів у послідовності бітів, що підвищує ефективність передачі даних. $\pi/4$ -DQPSK (differential quadrature phase shift keying) є комбінацією DQPSK і $\pi/4$ -QPSK, яка забезпечує хорошу стійкість до змін фази та вищу спектральну ефективність [1].

CPM - це клас методів, що використовують неперервну фазову модуляцію для кодування даних. Вона має високу спектральну ефективність та стійкість до шуму. MSK - це спеціальний випадок CPM зі зміщенням частоти на половину рідини символу. Вона має добру спектральну ефективність та добру стійкість до шуму. GMSK є модифікацією MSK з використанням гаусівського фільтра для зменшення спектральних бокових долів. Вона дозволяє отримати ще кращу спектральну ефективність. FSK використовує два або більше частотних станів для кодування даних. Вона має просту реалізацію, але меншу спектральну ефективність порівняно зі способами, які використовують фазову модуляцію.

Кожен з цих методів має свої переваги та недоліки і може бути оптимальним для певних застосувань залежно від вимог до спектральної ефективності, стійкості до шуму та складності реалізації. Для оцінки їх продуктивності та спектральної ефективності використовуються методи моделювання смуги пропускання та аналізу.

Модель смуги пропускання і еквівалентна модель основної смуги є двома основними методами моделювання каналу передачі даних. Розглянемо кожен із них.

В моделі смуги пропускання (імітаційна модель рівня сигналу) весь канал передачі даних (включаючи шум, затухання та інші ефекти) представляється як фільтр, який діє на сигнал. Зазвичай цей фільтр описується в частотній області, відображаючи вплив каналу на різні частоти. Ця модель дозволяє враховувати вплив різних фізичних ефектів на сигнал, таких як шум, затухання та спотворення, але вона може бути складною для обчислення, особливо для складних каналів.

В еквівалентній моделі основної смуги канал передачі даних абстрагується як ідеальний фільтр з певною передавальною функцією. Ця передавальна функція може бути виражена у вигляді складної функції часу або частоти. Така модель спрощує обчислення і дозволяє швидко оцінювати вплив каналу на сигнал, але вона може бути менш точною, оскільки враховується лише загальна характеристика каналу, а не його фактичний вплив на кожен сигнал.

Обидві ці моделі мають свої переваги та недоліки і використовуються в залежності від конкретних потреб моделювання та досліджень каналу передачі даних.

В симуляціях рівня сигналу для цифрової модуляції часто використовуються багато відліків на символ, щоб забезпечити достатню точність в оцінці різних характеристик сигналу. Проте, для обчислення частоти помилок можна скористатися лише значенням символу в момент часу дискретизації символу, не розглядаючи всю форму сигналу. У такому випадку еквівалентна дискретно-часова модель базової смуги може бути використана для отримання

результатів за короткий проміжок часу. Ця модель вимагає лише одну вибірку на символ, що дозволяє значно зменшити обсяг пам'яті, необхідний для симуляції, і забезпечує швидше обчислення. Такий підхід дозволяє ефективно використовувати ресурси обчислювального середовища для виконання симуляцій при високій точності оцінки помилок [2].

Несуча частота є основним сигналом, на який накладається інформація шляхом модуляції. Під час модуляції сигнал інформації (наприклад, аудіосигнал або дані) змінює параметри несучої хвилі, такі як амплітуда, частота або фаза, залежно від передаваної інформації. Наприклад, у FM-радіо амплітуда несучої частоти залишається постійною, але частота змінюється у відповідності зі змінами в аудіосигналі, що передається. У цьому випадку інформація про аудіо закодована відхиленнями частоти несучої хвилі.

У бездротових мережах центральна частота передачі може бути значно вищою, наприклад, у діапазоні гігагерців. Але навіть у цьому випадку, несуча частота залишається основним сигналом, на який накладається інформація для передачі.

Два основних типи моделей для моделювання систем зв'язку: модель смуги пропускання і її еквівалент в основній смузі. Модель смуги пропускання докладно моделює кожен цикл радіочастотної несучої, а спектр потужності сфокусований навколо несучої частоти. Ця модель вимагає багато пам'яті, оскільки кожна точка радіочастотної несучої повинна бути збережена для моделювання. З іншого боку, еквівалентна модель в основній смузі спрощує процес, замінюючи радіочастотну смугу на послідовність дискретних символів у часі. Ця модель споживає менше пам'яті, оскільки вона працює з відомими дискретними символами в часі, а не з високочастотними сигналами в смузі пропускання.

Вибір між цими двома моделями залежить від конкретних потреб і обмежень моделювання. Якщо необхідно докладно дослідити характеристики радіочастотної смуги, то модель смуги пропускання може бути більш відповідною. Однак, якщо ціллю є ефективне використання ресурсів

обчислювальної системи або якщо не потрібно детально моделювати радіочастотну поведінку, еквівалентна модель в основній смузі може бути більш практичним вибором.

У моделях основної смуги сигнали зосереджені навколо нульової частоти, і радіочастотна несуча пригнічується. Це дозволяє значно зменшити кількість відліків, необхідних для моделювання, оскільки лише область близької до нульової частоти є значущою для передачі інформації. Якщо поведінка системи вже добре вивчена і розуміється, модель основної смуги може бути подальше спрощена шляхом реалізації системи повністю на основі відліків у момент часу дискретизації символів. Це дозволяє ще більше зменшити обсяг обчислювальних ресурсів, які потрібні для моделювання, і полегшити розуміння поведінки системи [3].

Отже, обрання моделі для моделювання залежить від конкретних потреб і обмежень, а також від рівня розуміння поведінки системи та потреби у точності моделювання. В деяких випадках модель основної смуги може бути більш практичним вибором, особливо якщо потрібно ефективно використовувати ресурси обчислювальної системи і спростити моделювання.

2.2 Бінарна фазова маніпуляція (BPSK)

В схемі модуляції BPSK використовує два можливі фазові стани несучого сигналу для представлення бітів у двійковому повідомленні: один стан відповідає логічному "0", а інший - логічному "1". Основною ідеєю BPSK є використання фазового зміщення для кодування бітів. Наприклад, при використанні синусоїдального несучого сигналу, фаза сигналу змінюється на 180 градусів (або на протилежний фазовий стан) для кожного біту в залежності від його значення (0 або 1). Це дозволяє легко відокремлювати сигнали приймачем і спрощує декодування. Що стосується базисних функцій у методах цифрової модуляції, то вони дійсно часто обираються ортогональними одна до одної, що сприяє ефективному використанню спектру і спрощує демодуляцію. Процедура

ортогоналізації Грама-Шмідта може бути застосована для отримання таких базисних функцій. У випадку BPSK базисна функція є простою синусоїдою зі змінною фазою, оскільки вона дозволяє представити два можливих стани сигналу (0 та 1) шляхом зміни фази. Це дозволяє просто та ефективно використовувати частотний спектр і спрощує обробку сигналу як на стороні передавача, так і на стороні приймача [4].

На діаграмі сузір'їв для BPSK (рисунок 2.1) показано дві точки сузір'я, які повністю лежать на осі x (синфазні). Вони не мають проекції на вісь y (квадратура). Це означає, що сигнал, модульований BPSK, матиме синфазну складову, але не матиме квадратурної складової. Це пояснюється тим, що він має лише одну базисну функцію.

Можна відзначити, що фази несучої знаходяться на відстані 180 один від одного і вона має постійну огинаючу. Фаза несучої містить всю інформацію, яка передається.

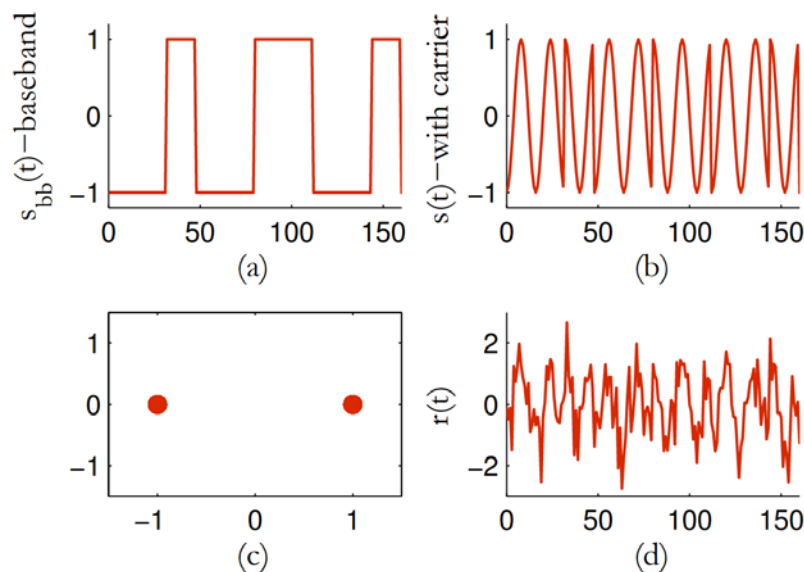


Рисунок 2.1 – (а) Сигнал BPSK базової смуги, (б) переданий сигнал BPSK - з несучою, (с) сузір'я на передавачі та (д) прийнятий сигнал з шумом AWGN

Наведемо приклад простої реалізації бінарної фазової маніпуляції (BPSK) на мові Python:

```
import numpy as np
```

```
def bpsk_modulation(data):
```

```
    """
```

Функція для модуляції даних методом бінарної фазової маніпуляції (BPSK).

Параметри:

- data: numpy array або список, що містить послідовність бітів для модуляції (0 та 1)

Повертає:

- modulated_signal: numpy array, що містить модульований сигнал BPSK

```
    """
```

```
    # Задаємо параметри сигналу
```

```
    frequency = 1000 # Частота несучої хвилі в Гц
```

```
    sampling_rate = 10000 # Частота дискретизації в Гц
```

```
    symbol_duration = 0.01 # Тривалість символу в секундах
```

```
    time = np.arange(0, len(data) * symbol_duration, 1 / sampling_rate)
```

```
    # Генеруємо несучий сигнал
```

```
    carrier_wave = np.sin(2 * np.pi * frequency * time)
```

```
    # Модуляція даних
```

```
    modulated_signal = np.array([])
```

```
    for bit in data:
```

```
        if bit == 0:
```

```
            modulated_signal = np.append(modulated_signal, carrier_wave * -1) #
```

Зміщення фази на 180 градусів

else:

```
modulated_signal = np.append(modulated_signal, carrier_wave)
```

```
return modulated_signal
```

Приклад використання

```
data = np.array([0, 1, 0, 1, 1, 0, 1]) # Послідовність бітів для модуляції (0 та 1)
modulated_signal = bpsk_modulation(data)
```

У цьому прикладі функція `'bpsk_modulation'` приймає послідовність бітів `'data'` і модулює їх за допомогою бінарної фазової маніпуляції (BPSK). Результатом є модульований сигнал BPSK, який зберігається у змінній `'modulated_signal'`.

2.3 Диференціальна модуляція BPSK (D-BPSK)

Розглянемо методи когерентного та некогерентного виявлення для сигналу Differential Binary Phase Shift Keying (DEBPSK), також відомого як Differential BPSK (DBPSK) або просто DPSK. DEBPSK кодує інформацію не як фазу самого сигналу, а як різницю фаз між двома послідовними відліками. Це означає, що кожен біт кодується не окремою фазою, а різницею фаз між поточним і попереднім відліками. Це дозволяє уникнути проблеми з некоректним виявленням фази сигналу, яка може виникнути через зміну початкової фази [5].

Метод когерентного виявлення використовується для виявлення DEBPSK, коли надається інформація про початкову фазу сигналу або коли можна точно визначити фазові зміни між послідовними відліками. Однак, якщо така інформація втрачається або недоступна, може бути використаний метод некогерентного виявлення. В некогерентному виявленні фазові зміни

вимірюються відносно попереднього вимірювання без врахування початкової фази.

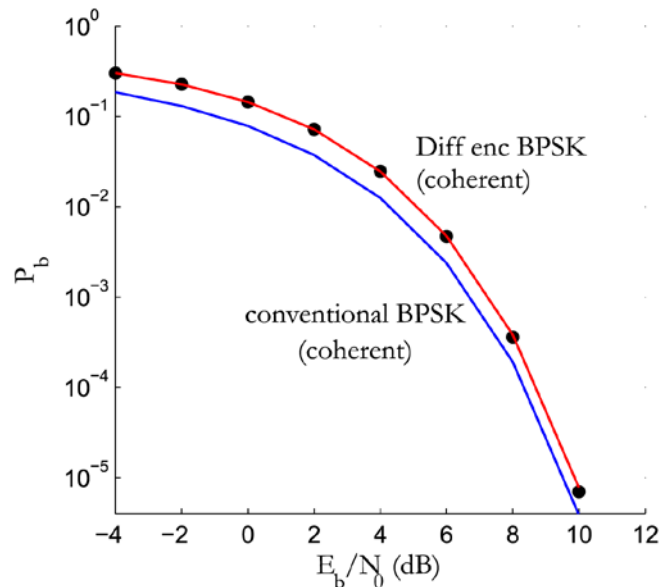


Рисунок 2.1 – Продуктивність диференційно кодованого BPSK та звичайного BPSK по каналу AWGN

Диференціальна демодуляція сигналу DEBPSK, яка іноді позначається як DBPSK або DPSK, зазвичай використовується для відновлення вихідних даних з диференційно кодованого сигналу. Цей метод дозволяє ефективно виявляти біти без потреби у точному відновленні початкової фази сигналу. У цілому, DEBPSK і його методи виявлення та демодуляції є важливими в сучасних системах зв'язку, особливо там, де проблема зміни початкової фази може виникнути через різні фізичні фактори, такі як затримки або спотворення сигналу [6].

Диференціальна модуляція BPSK (D-BPSK) кодує дані не як абсолютні фазові значення, але як зміни фази між послідовними символами. Сформуємо приклад реалізації D-BPSK на мові Python:

```
import numpy as np
```

```
def dbpsk_modulation(data):
```

```
"""
```

Функція для модуляції даних методом диференціальної бінарної фазової маніпуляції (D-BPSK).

Параметри:

- data: numpy array або список, що містить послідовність бітів для модуляції (0 та 1)

Повертає:

- modulated_signal: numpy array, що містить модульований сигнал D-BPSK

```
"""
```

```
# Задаємо параметри сигналу
```

```
frequency = 1000 # Частота несучої хвилі в Гц
```

```
sampling_rate = 10000 # Частота дискретизації в Гц
```

```
symbol_duration = 0.01 # Тривалість символу в секундах
```

```
time = np.arange(0, len(data) * symbol_duration, 1 / sampling_rate)
```

```
# Генеруємо несучий сигнал
```

```
carrier_wave = np.sin(2 * np.pi * frequency * time)
```

```
# Модуляція даних
```

```
modulated_signal = np.array([])
```

```
prev_phase = 0 # Попередня фаза
```

```
for bit in data:
```

```
    phase_shift = np.pi if bit == 1 else 0 # Зсув фази на 180 градусів для бітів
```

```
    phase = prev_phase + phase_shift # Розрахунок нової фази
```

```
    prev_phase = phase
```

```

modulated_symbol = np.sin(2 * np.pi * frequency * time + phase)
modulated_signal = np.append(modulated_signal, modulated_symbol)

return modulated_signal

# Приклад використання
data = np.array([0, 1, 0, 1, 1, 0, 1]) # Послідовність бітів для модуляції (0 та
1)
modulated_signal = dbpsk_modulation(data)

```

У цьому прикладі функція `dbpsk_modulation` приймає послідовність бітів `data` і модулює їх за допомогою диференціальної бінарної фазової маніпуляції (D-BPSK). Результатом є модульований сигнал D-BPSK, який зберігається у змінній `modulated_signal`.

2.4 Методи безперервної фазової модуляції телекомунікаційних сигналів

Безперервна фазова модуляція (BPSK) є однією з основних схем цифрової модуляції, де фаза несучого сигналу змінюється відповідно до вхідних бітів даних. У БФМ фаза сигналу може приймати будь-яке значення у визначеному діапазоні, що дозволяє кодувати більше інформації на символ порівняно з BPSK.

Одним із видів БФМ є квадратурна фазова модуляція (QPSK), де фазові стани розділяються на чотири можливі значення: 0° , 90° , 180° та 270° . Це дозволяє передавати два біта інформації за кожним символом, що робить QPSK більш ефективним з точки зору спектральної ефективності порівняно з BPSK. Іншими варіантами БФМ є 8-PSK, 16-PSK та інші, де фазовий простір поділений на більшу кількість можливих станів для кодування додаткової інформації на символ. Однак, збільшення кількості фазових станів може призвести до складності демодуляції та збільшення чутливості до шуму. Безперервна фазова модуляція використовується в багатьох системах зв'язку, таких як Wi-Fi,

Bluetooth, супутникове зв'язку, мобільні телефони та інші, де важлива ефективність спектра та стійкість до шуму [7].

Для зменшення впливу різких розривів фази в сигналі і мінімізації ефектів бічних пелюсток можна використовувати такі схеми модуляції, як квадратурна амплітудна модуляція (QAM) та фазово-маніпульована модуляція (PM). Ці схеми використовують плавні зміни амплітуди або фази несучої хвилі, що дозволяє підвищити спектральну ефективність та зменшити спотворення сигналу.

Квадратурна амплітудна модуляція (QAM) використовує як фазову, так і амплітудну компоненти несучої хвилі для передачі бітів даних. Ця схема дозволяє кодувати більше інформації у кожному символі, що дозволяє підвищити ефективність використання спектра та зменшити вплив бічних пелюсток.

Фазово-маніпульована модуляція (PM) змінює фазу несучої хвилі для передачі даних. У порівнянні зі звичайною PSK, де фаза різко змінюється на початку кожного символу, у PM фаза може змінюватися плавно від одного символу до іншого. Це дозволяє зменшити вплив бічних пелюсток та створити більш ефективний сигнал.

Використання цих схем модуляції допомагає зменшити вплив різких розривів фази в сигналі, підвищити ефективність використання спектра та зменшити спотворення сигналу, що є важливими факторами для забезпечення надійної передачі даних у радіозв'язку.

OQPSK є окремим випадком квадратурної фазової зміщеної модуляції (QPSK), де фаза сигналу змінюється лише на 90 градусів між сусідніми символами. Це обмеження діапазону зміни фази допомагає зменшити вплив бічних пелюсток. Проте, OQPSK не є схемою безперервної фазової модуляції (БФМ), оскільки фаза несучої не змінюється плавно від одного символу до іншого, а лише має фіксоване зміщення між символами. Для подальшого покращення рівнів бічних пелюсток та зменшення впливу різких розривів фази

можна використовувати схеми безперервної фазової модуляції, де фаза несучої плавно змінюється від одного символу до іншого в безперервному режимі.

Схеми безперервної фазової модуляції, такі як квадратурна фазова модуляція зі згладжуванням (CPM) або фазова маніпуляція зі згладжуванням (PM) дозволяють плавно змінювати фазу несучої хвилі, що дозволяє подолати обмеження OQPSK та зменшити вплив бічних пелюсток на сусідні канали та нелінійні ефекти підсилення [8].

Загальний лістинг реалізації безперервної фазової модуляції (БФМ) може залежати від конкретного використання та потреб системи телекомунікацій, а також від вибору конкретного алгоритму БФМ, такого як квадратурна фазова модуляція зі згладжуванням (CPM) або фазова маніпуляція зі згладжуванням (PM). Здійснимо реалізацію БФМ на прикладі квадратурної фазової модуляції зі згладжуванням (CPM) використовуючи мову програмування Python та бібліотеку NumPy.

```
import numpy as np

def continuous_phase_modulation(data, modulation_index, symbol_duration,
sampling_rate):
    """
```

Функція для реалізації безперервної фазової модуляції (БФМ) за допомогою квадратурної фазової модуляції зі згладжуванням (CPM).

Параметри:

- data: numpy array або список, що містить послідовність символів для модуляції
- modulation_index: коефіцієнт модуляції (зазвичай від 0 до 1)
- symbol_duration: тривалість одного символу в секундах
- sampling_rate: частота дискретизації сигналу в герцах

Повертає:

- modulated_signal: numpy array, що містить сигнал зі згладженим фазовим маніпуляціями

"""

Кількість відліків на символ

samples_per_symbol = int(symbol_duration * sampling_rate)

Створення порожнього масиву для зберігання згладжених фазових маніпуляцій

modulated_signal = np.array([])

Початкова фаза

phase = 0

Проходження по послідовності символів та модуляція

for symbol in data:

Зміна фази згідно зі значенням символу

phase += modulation_index * symbol

Зміна фази на кожному відліку

phases = np.linspace(phase - modulation_index * symbol, phase, samples_per_symbol, endpoint=False)

Обчислення комплексних чисел для згладження фазових маніпуляцій

complex_symbols = np.exp(1j * phases)

Додавання згладжених фазових маніпуляцій до сигналу

modulated_signal = np.append(modulated_signal, complex_symbols)

return modulated_signal

Приклад використання

```
data = np.array([0, 1, 0, 1, 1, 0, 1]) # Послідовність символів для модуляції (0
та 1)
modulation_index = 0.5 # Коефіцієнт модуляції
symbol_duration = 0.001 # Тривалість одного символу в секундах
sampling_rate = 10000 # Частота дискретизації сигналу в герцах
```

```
modulated_signal = continuous_phase_modulation(data, modulation_index,
symbol_duration, sampling_rate)
```

У цьому прикладі функція `continuous_phase_modulation` приймає послідовність символів `data`, коефіцієнт модуляції `modulation_index`, тривалість символу `symbol_duration` та частоту дискретизації `sampling_rate` і повертає модульований сигнал зі згладженою фазовою маніпуляцією.

2.5 Модуляція з безперервною фазовою маніпуляцією частоти (CPFSK)

Модуляція з безперервною фазовою маніпуляцією частоти (CPFSK) - це метод цифрової модуляції, в якому фаза несучої хвилі змінюється пропорційно зміні даних, а не як в BPSK або QPSK, де фаза змінюється миттєво при зміні даних. Наведемо приклад реалізації CPFSK на мові Python:

```
import numpy as np

def cpfsk_modulation(data, modulation_index, frequency_deviation,
symbol_duration, sampling_rate):
    """
```

Функція для модуляції даних методом безперервної фазової маніпуляції частоти (CPFSK).

Параметри:

- data: numpy array або список, що містить послідовність символів для модуляції

- modulation_index: коефіцієнт модуляції (зазвичай від 0 до 1)
- frequency_deviation: максимальний зсув частоти (в Гц)
- symbol_duration: тривалість одного символу в секундах
- sampling_rate: частота дискретизації сигналу в Гц

Повертає:

- modulated_signal: numpy array, що містить модульований сигнал CPFSK
 ""

Кількість відліків на символ

samples_per_symbol = int(symbol_duration * sampling_rate)

Створення порожнього масиву для зберігання модульованого сигналу

modulated_signal = np.array([])

Початкова фаза

phase = 0

Проходження по послідовності символів та модуляція

for symbol in data:

 # Зміна фази згідно зі значенням символу

 phase_shift = 2 * np.pi * modulation_index * symbol

 # Зміна фази на кожному відліку

 phases = np.linspace(phase, phase + phase_shift, samples_per_symbol,
 endpoint=False)

 # Обчислення комплексних чисел для модуляції

 complex_symbols = np.exp(1j * phases)

 # Додавання модульованих символів до сигналу

```

modulated_signal = np.append(modulated_signal, complex_symbols)

# Оновлення фази
phase += phase_shift

# Масштабування частоти сигналу
modulated_signal *= np.exp(1j * 2 * np.pi * frequency_deviation *
np.arange(len(modulated_signal)) / sampling_rate)

return modulated_signal

# Приклад використання
data = np.array([0, 1, 0, 1, 1, 0, 1]) # Послідовність символів для модуляції
modulation_index = 0.5 # Коефіцієнт модуляції
frequency_deviation = 5000 # Максимальний зсув частоти в Гц
symbol_duration = 0.001 # Тривалість одного символу в секундах
sampling_rate = 10000 # Частота дискретизації сигналу в Гц

modulated_signal = cpfsk_modulation(data, modulation_index,
frequency_deviation, symbol_duration, sampling_rate)

```

У цьому прикладі функція `cpfsk_modulation` приймає послідовність символів `data`, коефіцієнт модуляції `modulation_index`, максимальний зсув частоти `frequency_deviation`, тривалість символу `symbol_duration` та частоту дискретизації `sampling_rate`, і повертає модульований сигнал CPFSK.

Так, безперервна фазова маніпуляція з частотною маніпуляцією (CPFSK) є вдосконаленою версією технології частотної маніпуляції (FSK), яка усуває фазові розриви. У звичайній FSK-модуляції миттєва несуча частота перемикається між двома частотами. Однак через різке перемикання між цими двома частотами фаза в звичайній FSK не є безперервною.

У CPFSK фаза несучої хвилі плавно змінюється від одного символу до іншого, зберігаючи при цьому сталу частоту несучої хвилі. Це дозволяє уникнути різких розривів фази, що характерні для звичайної FSK, і полегшує демодуляцію сигналу. Одним із різновидів CPFSK є мінімально-зміщена безперервна фазова маніпуляція (MSK), де зміщення частоти між символами дорівнює половині ширини символу, що робить її особливо ефективною для систем з обмеженою шириною смуги.

Наступний фрагмент коду моделює двійкову CPFSK модуляцію ($h = 1$), а отримані осцилограми показано на рисунку 2.2. Фазова траєкторія, зображена на рисунку 2.2-(b), показує неоднозначність фазових переходів (фазовий перехід $+\pi$ еквівалентний переходу $-\pi$ за модулем арифметики 2π). Через це приймач не зможе використати фазову інформацію в двійковій CPFSK [9].

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import lfilter

L = 50 # oversampling factor
Tb = 0.5 # bit period in seconds
fs = L/Tb # sampling frequency in Hertz
fc = 2/Tb # carrier frequency
N = 8 # number of bits to transmit
h = 1 # modulation index

b = 2*np.random.randint(2, size=N)-1 # random information sequence in +1/-1 format
b = np.tile(b, (L,1)).flatten('F')
b_integrated = lfilter([1.0],[1.0,-1.0],b)/fs #Integrate b using filter

theta= np.pi*h/Tb*b_integrated
t=np.arange(0, Tb*N, 1/fs) # time base

s = np.cos(2*np.pi*fc*t + theta) # CPFSK signal

fig, (ax1,ax2,ax3) = plt.subplots(3, 1)

ax1.plot(t,b);ax1.set_xlabel('t');ax1.set_ylabel('b(t)')
ax2.plot(t,theta);ax2.set_xlabel('t');ax2.set_ylabel('$\theta(t)$')
ax3.plot(t,s);ax3.set_xlabel('t');ax3.set_ylabel('s(t)')
fig.show()
```

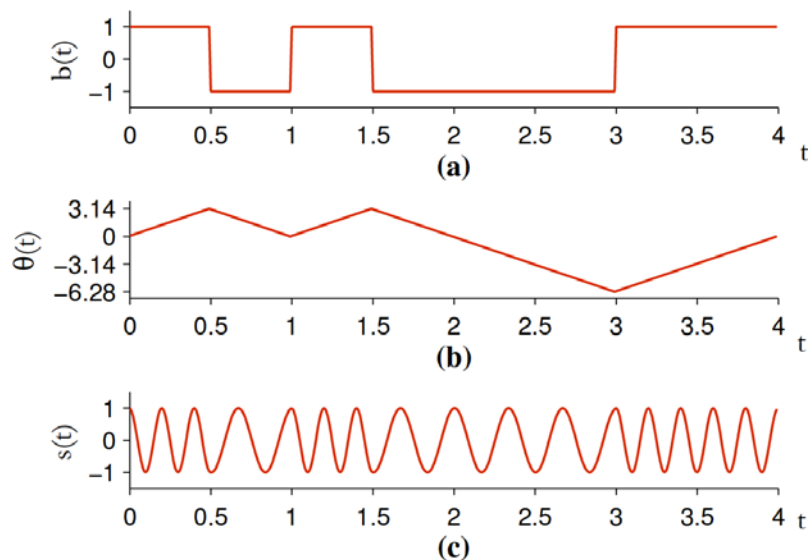


Рисунок 2.2 - (a) Інформаційна послідовність, (b) фазова характеристика CPFSK сигналу та (c) CPFSK модульований сигнал

2.6 Технологія мінімальної частотної маніпуляції

Техніка мінімальної маніпуляції (Continuous Phase Frequency Shift Keying, CPFSK) є формою мінімальної частотної маніпуляції, але частіше вона використовується у відомій версії, яка називається Minimum Shift Keying (MSK). MSK є підтипом CPFSK, де зміна частоти відбувається в строго визначених межах, що дозволяє досягнути деяких переваг, зокрема в MSK амплітуда сигналу залишається постійною, що робить його більш стійким до амплітудних спотворень в каналі зв'язку. Спектр MSK сигналу дуже ефективно використовує спектральні ресурси, що робить його ідеальним для застосувань з обмеженими спектральними ресурсами. MSK забезпечує добру стійкість до шуму і спотворень, що дозволяє високу швидкість передачі даних з низьким рівнем помилок [10].

У порівнянні з іншими методами модуляції, такими як Quadrature Phase Shift Keying (QPSK) або Offset Quadrature Phase Shift Keying (OQPSK), MSK може бути більш ефективним в деяких випадках, особливо коли важлива ефективність спектру та стійкість до спотворень.

MSK можна розглядати як окремий випадок двійкової CPFSK, де індекс модуляції h дорівнює 0,5. Той самий фрагмент коду, наведений для моделювання CPFSK, виконано з h , встановленим у 0,5, і результати показано на рисунку 2.3. Фазова траєкторія MSK на рисунку 2.3-(b), показує, що кожен інформаційний біт призводить до різних фазових переходів за модулем- 2π . Приймач може використовувати ці фазові переходи без будь-якої двозначності, і це може забезпечити кращі показники частоти помилок. Це основна мотивація, що лежить в основі методу MSK [11].

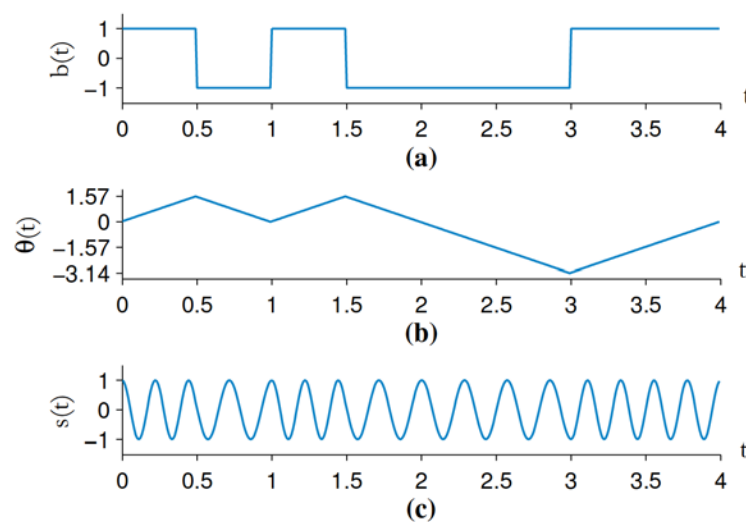


Рисунок 2.3 - (a) Інформаційна послідовність, (b) фазова характеристика MSK сигналу та (c) MSK модульований сигнал

2.7 Еквівалентність модуляції OQPSK

Модуляція Offset Quadrature Phase Shift Keying (OQPSK) є формою фазової модуляції, де кожен символ даних модулюється на двох неперекриваючих частотах, зсунутих по фазі на 90 градусів (або пів періоду). Це робить OQPSK ефективним для мінімізації впливу міжсимвольних інтерференцій (ISI) в каналі передачі даних. Модуляція OQPSK може бути еквівалентною модуляції MSK (Minimum Shift Keying) у певних умовах. Зокрема, якщо вхідний потік даних

MSK складається з послідовності бітів, де кожний символ даних відповідає двом послідовним бітам, то MSK і OQPSK можуть бути еквівалентними. В MSK кожен символ даних кодується зміною фази сигналу відносно попереднього символу, а в OQPSK кожен символ даних кодується як зміна фази і амплітуди відносно попереднього символу. Отже, якщо вхідні дані MSK можуть бути перетворені на послідовність, яку можна модулювати за допомогою OQPSK, то ці дві модуляції можуть бути еквівалентними в певних умовах. Однак в реальних застосуваннях вибір між MSK та OQPSK може залежати від конкретних вимог до системи зв'язку та характеристик каналу передачі даних.

Рисунок 2.4 ілюструє схожість побудови сигналів OQPSK і MSK. Так, і OQPSK (Offset Quadrature Phase Shift Keying), і MSK (Minimum Shift Keying) є формами фазової модуляції, і вони мають деякі схожі риси у побудові сигналів [12].

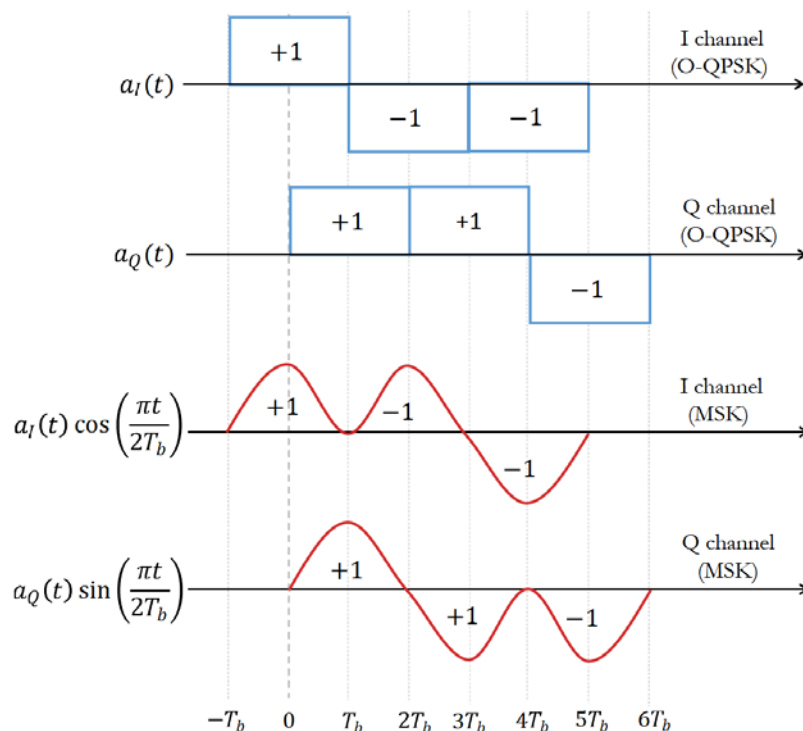


Рисунок 2.4 - Подібність між формами сигналів OQPSK та MSK

У OQPSK сигнал будується шляхом розділення вхідного потоку даних на два потоки: синфазний (In-Phase, I) і квадратурний (Quadrature, Q). Кожен з цих потоків модулюється за допомогою фазового зсуву на 90 градусів (або половини періоду символу) відносно попереднього символу. Це зменшує ефект міжсимвольних інтерференцій (ISI) і робить сигнал більш стійким до спотворень. У MSK також є дві складові: синфазна та квадратурна. Вони також зсунуті на половину періоду символу відносно попереднього символу, але додатково формуються функціями косинуса і синуса, які мають напівциклічну форму. Ця форма модуляції дозволяє забезпечити постійну огибаючу сигналу, що робить MSK більш ефективним у використанні спектральних ресурсів.

Отже, якщо дивитися на спектральний склад сигналів OQPSK і MSK, то вони можуть бути дуже схожими, оскільки обидва методи спрямовані на мінімізацію спектральної ширини сигналу і забезпечення ефективного використання спектральних ресурсів. Однак в деталях побудови сигналу і способу формування фазових компонентів є нюанси, які роблять ці методи трохи відмінними один від одного [1].

2.8 Дослідження властивостей фазових переходів

Властивості фазового переходу різних варіантів схем QPSK і MSK можна легко дослідити за допомогою графіків сузір'їв. Наступний приклад коду демонструє, як побудувати графік сузір'їв сигнального простору для різних модуляцій таких як QPSK і MSK, що використовуються в передавачі на мові Python.

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

# Функція для побудови сузір'їв QPSK
def plot_qpsk_constellation():
```

```
symbols = np.array([1 + 1j, 1 - 1j, -1 + 1j, -1 - 1j]) / np.sqrt(2) # Символи
QPSK
```

```
plt.scatter(np.real(symbols), np.imag(symbols), marker='o', color='b') #
Побудова графіка сузір'я
```

```
plt.title('QPSK Constellation')
plt.xlabel('I (In-phase)')
plt.ylabel('Q (Quadrature)')
plt.grid(True)
plt.axis('equal')
plt.show()
```

```
# Функція для побудови сузір'їв MSK
```

```
def plot_msk_constellation():
    symbols = np.array([np.cos(np.pi / 4) + 1j * np.sin(np.pi / 4), np.cos(3 * np.pi
/ 4) + 1j * np.sin(3 * np.pi / 4)]) # Символи MSK
    plt.scatter(np.real(symbols), np.imag(symbols), marker='o', color='r') #
```

```
Побудова графіка сузір'я
```

```
plt.title('MSK Constellation')
plt.xlabel('I (In-phase)')
plt.ylabel('Q (Quadrature)')
plt.grid(True)
plt.axis('equal')
plt.show()
```

```
# Виклик функцій для побудови сузір'їв
```

```
plot_qpsk_constellation()
plot_msk_constellation()
```

У цьому коді функція `plot_qpsk_constellation()` побудовує графік сузір'я QPSK, а функція `plot_msk_constellation()` - графік сузір'я MSK. Обидва графіки

використовують вбудовану функцію `scatter` бібліотеки Matplotlib для відображення точок сузір'я на комплексній площині.

Зазвичай, у практичному застосуванні, модульовані сигнали в основній смузі пропускаються через фільтр формування імпульсів для боротьби з явищем міжсимвольної інтерференції (ISI). Метою роботи є побудова графіків сузір'я різних імпульсних форм сигналів основної смуги частот для схем QPSK, O-QPSK і $\pi/4$ -DQPSK. Існує безліч фільтрів для формування імпульсів, і для цієї демонстрації спеціально вибрано фільтр з піднесенням косинусом.

Імпульс підвищеного косинуса (RC) має регульований параметр зсуву перехідної смуги α , за допомогою якого можна контролювати затухання перехідної смуги.

Наведемо приклад коду на мові Python, який демонструє побудову графіків сузір'їв для різних модуляцій (QPSK, O-QPSK та $\pi/4$ -DQPSK) з використанням імпульсів підвищеного косинуса (RC):

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import rcosfilter
# Функція для побудови сузір'я з імпульсами підвищеного косинуса
def plot_rcos_constellation(alpha):
    # Генерація фільтра підвищеного косинуса
    ntaps = 101
    beta = alpha
    taps = rcosfilter(ntaps, alpha, 1, 1)

    # Побудова сузір'я
    symbols = np.random.choice([-1, 1], size=1000) + 1j * np.random.choice([-1, 1], size=1000)
    filtered_symbols = np.convolve(symbols, taps[1], mode='same')
```

```
plt.scatter(np.real(filtered_symbols), np.imag(filtered_symbols), marker='o',
color='b')

plt.title('RCOS Constellation (alpha={}).format(alpha))
plt.xlabel('I (In-phase)')
plt.ylabel('Q (Quadrature)')
plt.grid(True)
plt.axis('equal')
plt.show()

# Виклик функції для побудови сузір'я з різними значеннями параметра
alpha
alphas = [0.5, 0.75, 1.0] # Різні значення параметра alpha для імпульсу
підвищеного косинуса
for alpha in alphas:
    plot_rcos_constellation(alpha)
```

У цьому коді функція `plot_rcos_constellation(alpha)` генерує сузір'я для заданого значення параметра `alpha`, що визначає форму імпульсу підвищеного косинуса. Для кожного значення `alpha` генерується випадкова послідовність символів QPSK, яка фільтрується за допомогою фільтра підвищеного косинуса, а потім відображається на комплексній площині. Виконаємо реалізацію лістингу формувача імпульсів підвищеного косинуса.

```
def raisedCosineDesign(alpha, span, L):
    """
    Raised cosine FIR filter design
    Parameters:
        alpha : roll-off factor
        span : filter span in symbols
        L : oversampling factor (i.e, each symbol contains L samples)
    Returns:
        p - filter coefficients b of the designed
            FIR raised cosine filter
    """
    t = np.arange(-span/2, span/2 + 1/L, 1/L) # +/- discrete-time base
    with np.errstate(divide='ignore', invalid='ignore'):
        A = np.divide(np.sin(np.pi*t), (np.pi*t)) #assume Tsym=1
        B = np.divide(np.cos(np.pi*alpha*t), 1-(2*alpha*t)**2)
        p = A*B
    #Handle singularities
    p[np.argwhere(np.isnan(p))] = 1 # singularity at p(t=0)
    # singularity at t = +/- Tsym/2alpha
    p[np.argwhere(np.isinf(p))] = (alpha/2)*np.sin(np.divide(np.pi, (2*alpha)))
    return p
```

Функція тестується за допомогою наступного коду. Вона генерує підвищений косинусоїдальний імпульс для заданої тривалості символу $T_{\text{sym}} = 1$ с і будує часову діаграму та частотну характеристику, як показано на рисунку 2.5.

Наведемо лістинг для формування піднятих косинусоїдальних імпульсів в часовій та частотній області.

```
def raisedCosineDemo():
    """
    Raised Cosine pulses and their manifestation in frequency domain
    Usage:
        >> from DigiCommPy.pulseshapers import raisedCosineDemo
        >> raisedCosineDemo()
    """
    import matplotlib.pyplot as plt
    from scipy.fftpack import fft, fftshift

    Tsym = 1 # Symbol duration in seconds
    L = 32 # oversampling rate, each symbol contains L samples
    span = 10 # filter span in symbols
    alphas= [0, 0.3, 0.5, 1] # RC roll-off factors - valid range 0 to 1
    Fs = L/Tsym # sampling frequency

    lineColors = ['b', 'r', 'g', 'k']
    fig, (ax1, ax2) = plt.subplots(1, 2)
```

```

for i, alpha in enumerate(alphas):
    b = raisedCosineDesign(alpha,span,L) # RC Pulse design

    # time base for symbol duration
    t = Tsym* np.arange(-span/2, span/2 + 1/L, 1/L)

    # plot time domain view
    ax1.plot(t,b,lineColors[i],label=r'$\alpha$='+str(alpha))

    #Compute FFT and plot double sided frequency domain view
    NFFT = 1<<(len(b)-1).bit_length() #Set FFT length = nextpower2(len(b))
    vals = fftshift(fft(b,NFFT))
    freqs = Fs* np.arange(-NFFT/2,NFFT/2)/NFFT
    ax2.plot(freqs,abs(vals)/abs(vals[len(vals)//2]),\
              lineColors[i],label=r'$\alpha$='+str(alpha))

ax1.set_title('Raised cosine pulse');
ax2.set_title('Frequency response')
ax1.legend();ax2.legend()
fig.show()

```

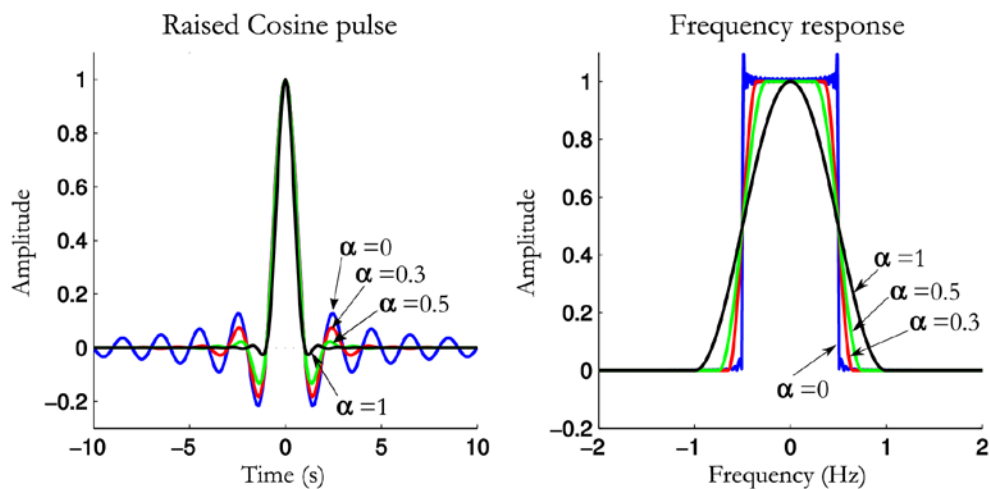


Рисунок 2.5 - Підвищений косинусоїдальний імпульс та його відображення у частотній області

Наведемо приклад коду, включаючи функцію для генерації модульованих сигналів з використанням схем QPSK, O-QPSK та p=4-DQPSK, а також їхню фільтрацію через фільтр підвищеного косинуса з коефіцієнтом згортання $\alpha = 0.3$:

```

import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import rcosfilter

# Функція для генерації фільтра підвищеного косинуса
def generate_rcos_filter(alpha):
    ntaps = 101
    beta = alpha
    taps = rcosfilter(ntaps, alpha, 1, 1)
    return taps[1]

# Функція для генерації сигналу з модуляцією QPSK
def generate_qpsk_signal(num_symbols):
    symbols = np.random.choice([-1, 1], size=num_symbols) + 1j *
np.random.choice([-1, 1], size=num_symbols)
    return symbols

# Функція для модуляції сигналу O-QPSK
def modulate_oqpsk_signal(symbols):
    even_symbols = symbols[::2]
    odd_symbols = symbols[1::2]
    return np.concatenate([even_symbols, 1j * odd_symbols])

# Функція для генерації сигналу з модуляцією p=4-DQPSK
def generate_p4_dqpsk_signal(num_symbols):
    initial_symbol = np.random.choice([-1, 1]) + 1j * np.random.choice([-1, 1])
    symbols = [initial_symbol]
    for _ in range(1, num_symbols):
        symbol = np.random.choice([-1, 1]) + 1j * np.random.choice([-1, 1])
        symbols.append(np.exp(1j * np.angle(symbol) + 1j * np.pi / 4))

```

```

return np.array(symbols)

# Функція для модуляції сигналу MSK
def modulate_msk_signal(symbols):
    return np.exp(1j * np.pi / 4) * symbols

# Функція для фільтрації сигналу через фільтр підвищеного косинуса
def filter_signal(signal, filter_taps):
    return np.convolve(signal, filter_taps, mode='same')

# Функція для побудови сузір'я сигналу
def plot_constellation(signal, title):
    plt.scatter(np.real(signal), np.imag(signal), marker='o', color='b')
    plt.title(title)
    plt.xlabel('I (In-phase)')
    plt.ylabel('Q (Quadrature)')
    plt.grid(True)
    plt.axis('equal')
    plt.show()

# Генерація фільтра підвищеного косинуса з коефіцієнтом згортання  $\alpha = 0.3$ 
alpha = 0.3
rcos_filter = generate_rcos_filter(alpha)

# Генерація сигналів та їх модуляція
num_symbols = 1000
qpsk_signal = generate_qpsk_signal(num_symbols)
oqpsk_signal = modulate_oqpsk_signal(qpsk_signal)
p4_dqpsk_signal = generate_p4_dqpsk_signal(num_symbols)
msk_signal = modulate_msk_signal(qpsk_signal)

```

```

# Фільтрація сигналів через фільтр підвищеного косинуса
filtered_qpsk_signal = filter_signal(qpsk_signal, rcos_filter)
filtered_oqpsk_signal = filter_signal(oqpsk_signal, rcos_filter)
filtered_p4_dqpsk_signal = filter_signal(p4_dqpsk_signal, rcos_filter)
filtered_msk_signal = filter_signal(msk_signal, rcos_filter)

# Побудова сузір'їв сигналів
plot_constellation(filtered_qpsk_signal, 'QPSK Constellation after RC Filter ( $\alpha = 0.3$ )')
plot_constellation(filtered_oqpsk_signal, 'O-QPSK Constellation after RC Filter ( $\alpha = 0.3$ )')
plot_constellation(filtered_p4_dqpsk_signal, 'p=4-DQPSK Constellation after RC Filter ( $\alpha = 0.3$ )')
plot_constellation(filtered_msk_signal, 'MSK Constellation after RC Filter ( $\alpha = 0.3$ )')

```

У цьому коді спочатку генеруються фільтр підвищеного косинуса з коефіцієнтом згортання $\alpha = 0.3$, а потім генеруються та модулюються сигнали QPSK, O-QPSK, $\pi/4$ -DQPSK і MSK. Потім кожен сигнал фільтрується через фільтр підвищеного косинуса, і, нарешті, побудовано сузір'я кожного з фільтрованих сигналів.

Наведемо лістинг формування діаграми сузір'я RC-фільтрованих QPSK та MSK.

```

import numpy as np
import matplotlib.pyplot as plt
from DigiCommPy.passband_modulations import
    → qpsk_mod, oqpsk_mod, piBy4_dqpsk_mod, msk_mod
from DigiCommPy.pulsesapers import raisedCosineDesign

N=1000 # Number of symbols to transmit, keep it small and adequate
fc=10; L=8 # carrier frequency and oversampling factor
a = np.random.randint(2, size=N) # uniform random symbols from 0's and 1's

#modulate the source symbols using QPSK, QPSK, pi/4-DQPSK and MSK
qpsk_result= qpsk_mod(a,fc,L)
oqpsk_result = oqpsk_mod(a,fc,L)
piBy4qpsk_result = piBy4_dqpsk_mod(a,fc,L)
msk_result = msk_mod(a,fc,L);

#Pulse shape the modulated waveforms by convolving with RC filter
alpha = 0.3; span = 10 # RC filter alpha and filter span in symbols
b =raisedCosineDesign(alpha,span, L) # RC pulse shaper
iRC_qpsk=np.convolve(qpsk_result['I(t)'],b,mode='valid')#RC - QPSK I(t)
qRC_qpsk= np.convolve(qpsk_result['Q(t)'],b,mode='valid') #RC - QPSK Q(t)
iRC_oqpsk= np.convolve(oqpsk_result['I(t)'],b,mode='valid')#RC - OQPSK I(t)
qRC_oqpsk= np.convolve(oqpsk_result['Q(t)'],b,mode='valid')#RC - OQPSK Q(t)
iRC_piBy4qpsk=np.convolve(piBy4qpsk_result['U(t)'],b,mode='valid')#pi/4-QPSK I
qRC_piBy4qpsk=np.convolve(piBy4qpsk_result['V(t)'],b,mode='valid')#pi/4-QPSK Q
i_msk = msk_result['sI(t)'] # MSK sI(t)
q_msk = msk_result['sQ(t)'] # MSK sQ(t)

fig, axs = plt.subplots(2, 2)

axs[0,0].plot(iRC_qpsk,qRC_qpsk)# RC shaped QPSK
axs[0,1].plot(iRC_oqpsk,qRC_oqpsk)# RC shaped OQPSK
axs[1,0].plot(iRC_piBy4qpsk,qRC_piBy4qpsk)# RC shaped pi/4-QPSK
axs[1,1].plot(i_msk[20:-20],q_msk[20:-20])# RC shaped OQPSK

axs[0,0].set_title(r'QPSK, RC $\alpha$='+str(alpha))
axs[0,0].set_xlabel('I(t)');axs[0,0].set_ylabel('Q(t)');
axs[0,1].set_title(r'OQPSK, RC $\alpha$='+str(alpha))
axs[0,1].set_xlabel('I(t)');axs[0,1].set_ylabel('Q(t)');
axs[1,0].set_title(r'$\pi/4$ - QPSK, RC $\alpha$='+str(alpha))
axs[1,0].set_xlabel('I(t)');axs[1,0].set_ylabel('Q(t)');
axs[1,1].set_title('MSK')
axs[1,1].set_xlabel('I(t)');axs[1,1].set_ylabel('Q(t)');
fig.show()

```

Отриманий в результаті моделювання графік зображено на рисунку 2.6. З отриманого графіка сузір'я можна зробити наступні висновки.

- Звичайна QPSK має 180 фазових переходів і, отже, вимагає лінійних підсилювачів з високою добротністю
- Фазові переходи Offset-QPSK обмежуються 90 (180 фазових переходів усуваються)

- Сигнальні точки для $\pi/4$ -DQPSK перемикаються між двома наборами QPSK-сузір'їв, які зсунуті на 45° відносно один одного. У цьому сузір'ї відсутні фазові переходи 90° і 180° [2].

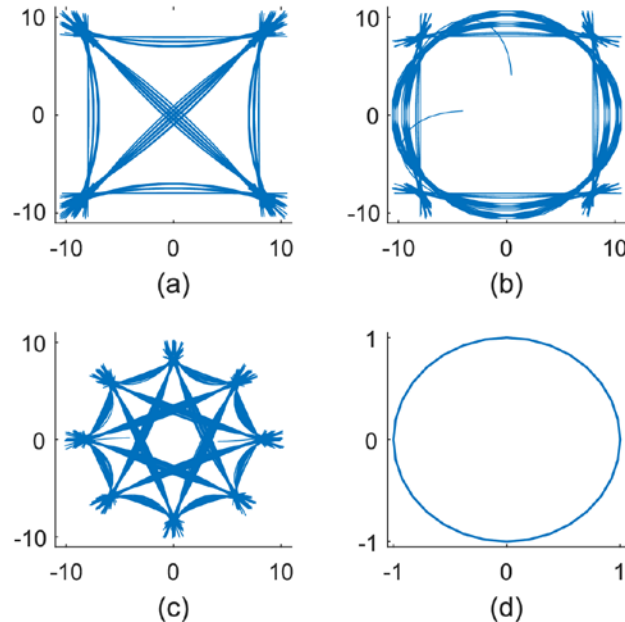


Рисунок 2.6 - Графіки сузір'їв для: (a) $\alpha = 0,3$ RC-заповнення QPSK, (b) $\alpha = 0,3$ RC-заповнення O-QPSK, (c) $\alpha = 0,3$ RC-заповнення $\pi/4$ -DQPSK та (d) MSK

Тому ця схема дає менші варіації огибаючої, ніж інші дві схеми QPSK. - MSK - це безперервна фазова модуляція, тому при зміні символу не відбувається різкого фазового переходу. На це вказує плавне коло на графіку сузір'я. Отже, сигнал з обмеженою смугою пропускання MSK не зазнає жодних змін огибаючої, тоді як решта схем QPSK страждають від різних рівнів змін огибаючої, коли вони обмежені смугою пропускання [3].

2.9 Висновки до розділу 2

На основі результатів дослідження методів модуляції в бездротових системах можна зробити наступні висновки. Різні методи модуляції мають різні характеристики, що впливають на ефективність передачі даних. Наприклад,

методи з квадратурною модуляцією, такі як QPSK (Quadrature Phase Shift Keying) та QAM, можуть бути менш вразливими до шуму та спотворень у порівнянні з PSK. Різні методи модуляції мають різні властивості щодо спектрального використання. Деякі методи, такі як MSK (Minimum Shift Keying), можуть забезпечувати більш компактний спектр, що дозволяє краще використовувати доступну спектральну ширину каналу. Різні методи модуляції можуть вимагати різного рівня складності апаратного забезпечення для реалізації. Наприклад, методи з квадратурною модуляцією можуть потребувати більш складного обладнання для демодуляції. Вибір конкретного методу модуляції залежить від вимог конкретної бездротової системи. Наприклад, системи з високою швидкістю передачі даних можуть вимагати використання методів з більшою кількістю бітів на символ, тоді як системи з вимогами до спектрального ефективного використання можуть вибирати методи з компактним спектром, такі як MSK. У цілому, вибір конкретного методу модуляції повинен бути здійснений з урахуванням специфіки конкретної бездротової системи, вимог до передачі даних, стійкості до шуму та перешкод, а також вимог до апаратного забезпечення.

3 СПЕКТРАЛЬНА ЩІЛЬНІСТЬ ПОТУЖНОСТІ МОДУЛЬОВАНИХ ТЕЛЕКОМУНІКАЦІЙНИХ СИГНАЛІВ

Спектральна щільність потужності (PSD) - це міра інтенсивності потужності сигналу в частотній області. Порівняння PSD різних схем модуляції відіграє важливу роль у розумінні їхніх спектральних характеристик і прийнятті рішень щодо вибору схеми модуляції, яка задовольняє вимогам конкретної системи зв'язку.

3.1 Методи оцінювання спектральної густини потужності сигналів

Спектральна щільність потужності (PSD) допомагає виміряти, як потужність сигналу розподілена по частоті. Однак, якщо сигнал є стохастичним за своєю природою, спроба обчислити спектральні компоненти сигналу за допомогою дискретного перетворення Фур'є (ДПФ) не буде правильною, оскільки для кожної реалізації випадкового процесу обчислений ДПФ буде змінюватися.

Для стохастичних сигналів, таких як шум, використання методів, що базуються на аналізі часової і частотної областей, є більш практичним. Наприклад, для оцінки PSD стохастичного сигналу часто використовують методи, такі як метод Уелча або метод Бартлетта, які базуються на усередненні кількох перетворень Фур'є. Метод Уелча розбиває сигнал на кілька неперекриваючих сегментів, обчислює ДПФ для кожного сегмента, а потім обчислює середнє значення квадрата абсолютної величини ДПФ для всіх сегментів. Це дозволяє отримати більш стабільну оцінку PSD для стохастичного сигналу. Метод Бартлетта використовує подібний підхід, але розділяє сигнал на перекриваючі сегменти, що дозволяє отримати ще більшу статистичну незалежність між вимірюваннями [4].

Ці методи є ефективними для оцінки PSD стохастичних сигналів та допомагають уникнути проблем, пов'язаних з варіаціями в обчисленнях ДПФ для кожної реалізації сигналу.

Теорема Вінера-Хінчина є потужним інструментом для аналізу стаціонарних випадкових процесів в широкому сенсі, які можуть бути описані їхніми автокореляційними функціями. Згідно з цією теоремою, спектральну щільність потужності (PSD) стаціонарного випадкового процесу можна обчислити як перетворення Фур'є його автокореляційної функції. Іншими словами, якщо у нас є достатньо точна модель автокореляційної функції стаціонарного випадкового процесу, ми можемо обчислити його PSD, застосовуючи перетворення Фур'є до цієї автокореляційної функції (рис. 3.1) [5].

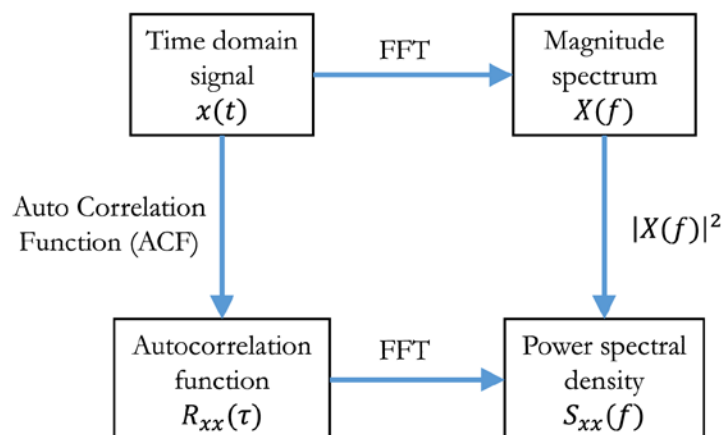


Рисунок 3.1 – Візуалізація теореми Вінера-Хінчина

Такий підхід дозволяє нам аналізувати спектральні властивості стаціонарних випадкових процесів і визначати розподіл енергії сигналу в частотній області. Це особливо корисно для розуміння спектральних характеристик сигналів і шумів у системах зв'язку, радіо, радіолокації, обробки сигналів та інших додатках, де аналіз частотного спектру є важливим [6].

Для аналізу стохастичного сигналу, якщо немає достатньої кількості статистичних даних (що часто трапляється на практиці), необхідно оцінити спектральну щільність потужності. Існує декілька методів оцінки PSD, одним з яких є метод Велша. Метод Велша оцінює спектр потужності сигналу шляхом розбиття його на блоки, що перекриваються, обчислення періодограми (перетворення Фур'є оцінки автокореляційної послідовності) та усереднення результатів. Так, метод Велша є одним із найпоширеніших методів оцінки спектральної щільності потужності (PSD) стохастичного сигналу. Цей метод дозволяє оцінювати PSD шляхом розбиття сигналу на блоки, обчислення періодограми (перетворення Фур'є оцінки автокореляційної послідовності) для кожного блоку та усереднення результатів [7].

У мові Python для оцінки PSD методом Велша можна використати функцію `welch` з пакета `scipy.signal`. Ця функція автоматично розбиває сигнал на блоки, обчислює періодограми для кожного блоку та усереднює їх, надаючи оцінку PSD. Наведемо приклад використання функції `welch` для оцінки PSD.

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import welch

# Згенеруємо стохастичний сигнал
np.random.seed(0)
n_samples = 1000
signal = np.random.randn(n_samples)

# Оцінка PSD методом Велша
frequencies, psd = welch(signal, nperseg=256)

# Побудова графіка PSD
plt.semilogy(frequencies, psd)
```

```
plt.title('PSD за методом Велша')
plt.xlabel('Частота')
plt.ylabel('PSD')
plt.grid(True)
plt.show()
```

У цьому прикладі функція `welch` використовується для обчислення PSD для стохастичного сигналу `signal`. Параметр `nperseg` вказує розмір кожного блоку при оцінці PSD. Результати оцінки PSD потім відображаються на графіку за допомогою `matplotlib.pyplot.semilogy`, який відображає PSD в логарифмічному масштабі по вертикалі.

Наступна функція легко будує графік оцінки спектра Велча за допомогою функції Велча з пакета SciPy. Функція використовує рекомендовані налаштування з [5], де функцію Велша налаштовано з розкладкою Ганнінга, без перекриття і з коефіцієнтом усереднення 16. Ця функція використовується для побудови оцінок PSD сигналів, модульованих різними методами цифрової модуляції, а саме: BPSK, QPSK, MSK і GMSK [8].

Наведемо загальний шаблон функції:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import welch
```

```
def plot_welch_spectrum(signal, fs, modulation_type):
```

```
    """
```

Функція для побудови оцінки спектра Велча для заданого сигналу.

Параметри:

`signal` (numpy.array): Вхідний сигнал.

`fs` (float): Частота дискретизації сигналу.

```

modulation_type (str): Тип модуляції ('BPSK', 'QPSK', 'MSK', 'GMSK').
"""

# Налаштування параметрів для оцінки спектра Велча
nperseg = 256 # Розмір вікна для оцінки PSD
average_method = 'mean' # Метод усереднення

# Рекомендовані параметри для оцінки спектра Велча
if average_method == 'mean':
    averaging_factor = 16 # Коефіцієнт усереднення

# Використання функції welch для оцінки PSD
frequencies, psd = welch(signal, fs=fs, nperseg=nperseg,
averaging=average_method, average=averaging_factor)

# Побудова графіка PSD
plt.semilogy(frequencies, psd)
plt.title(f'Welch Spectrum Estimate for {modulation_type}')
plt.xlabel('Frequency (Hz)')
plt.ylabel('PSD')
plt.grid(True)
plt.show()

# Приклад використання функції
# Замість signal та fs введіть ваші дані та частоту дискретизації.
# Замість 'BPSK' введіть тип вашої модуляції ('QPSK', 'MSK', 'GMSK').
# plot_welch_spectrum(signal, fs, 'BPSK')

```

Цей шаблон дозволяє викликати функцію `plot_welch_spectrum` із сигналом, частотою дискретизації та типом модуляції. Вона автоматично оцінить PSD за

допомогою методу Велча та побудує графік. При цьому, потрібно буде вставити власні дані сигналу та частоту дискретизації.

Отже, якщо ми маємо сигнал з модуляцією несучої, можемо спочатку створити такий сигнал і потім використати метод Велша для оцінки його спектральної щільності потужності (PSD) [9].

Для прикладу створимо сигнал з амплітудною модуляцією несучої та оцінимо його PSD за допомогою методу Велша. Після цього ми побудуємо графік PSD.

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import welch

# Параметри сигналу
fs = 1000 # Частота дискретизації
T = 1 # Тривалість сигналу
t = np.arange(0, T, 1/fs) # Вектор часу

# Створення несучої хвилі
carrier_freq = 10 # Частота несучої хвилі
carrier_signal = np.sin(2 * np.pi * carrier_freq * t)

# Створення модулюючого сигналу
modulating_freq = 1 # Частота модулюючого сигналу
modulating_signal = 0.5 * np.sin(2 * np.pi * modulating_freq * t)

# Амплітудна модуляція несучої хвилі модулюючим сигналом
modulated_signal = (1 + modulating_signal) * carrier_signal

# Оцінка PSD методом Велша
```

```
frequencies, psd = welch(modulated_signal, fs=fs)

# Побудова графіка PSD
plt.semilogy(frequencies, psd)
plt.title('PSD Велша для сигналу з модуляцією несучої')
plt.xlabel('Частота (Гц)')
plt.ylabel('PSD')
plt.grid(True)
plt.show()
```

У цьому прикладі ми створили несучу хвилю з частотою 10 Гц і модулюючий сигнал з частотою 1 Гц. За допомогою амплітудної модуляції ми створили модульований сигнал. Потім ми оцінили його PSD за допомогою методу Велша та побудували графік PSD.

```
def plotWelchPSD(x,fs,fc,ax = None,color='b', label=None):
    """
    Plot PSD of a carrier modulated signal using Welch estimate
    Parameters:
        x : signal vector (numpy array) for which the PSD is plotted
        fs : sampling Frequency
        fc : center carrier frequency of the signal
        ax : Matplotlib axes object reference for plotting
        color : color character (format string) for the plot
    """
    from scipy.signal import hanning, welch
    from numpy import log10
    nx = max(x.shape)
    na = 16 # averaging factor to plot averaged welch spectrum
    w = hanning(nx//na) #// is for integer floor division
    # Welch PSD estimate with Hanning window and no overlap
    f, Pxx = welch(x,fs>window = w,noverlap=0)
    indices = (f>=fc) & (f<4*fc) # To plot PSD from Fc to 4*Fc
    Pxx = Pxx[indices]/Pxx[indices][0] # normalized psd w.r.t Fc
    ax.plot(f[indices]-fc,10*log10(Pxx),color,label=label) #Plot in the given axes
```

3.2 Лістинги програмного визначення спектральної щільності потужності (PSD)

Наведемо лістинг побудови графіків спектральної щільності потужності (PSD) для кожного з фільтрованих сигналів. Для цього ми можемо використати функцію `matplotlib.pyplot.psd`, яка автоматично обчислює PSD і побудовує графік.

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import rcosfilter

# Функція для генерації фільтра підвищеного косинуса
def generate_rcos_filter(alpha):
    ntaps = 101
    beta = alpha
    taps = rcosfilter(ntaps, alpha, 1, 1)
    return taps[1]

# Функція для генерації сигналу з модуляцією QPSK
def generate_qpsk_signal(num_symbols):
    symbols = np.random.choice([-1, 1], size=num_symbols) + 1j *
np.random.choice([-1, 1], size=num_symbols)
    return symbols

# Функція для модуляції сигналу O-QPSK
def modulate_oqpsk_signal(symbols):
    even_symbols = symbols[::2]
    odd_symbols = symbols[1::2]
    return np.concatenate([even_symbols, 1j * odd_symbols])
```

Функція для генерації сигналу з модуляцією p=4-DQPSK

```
def generate_p4_dqpsk_signal(num_symbols):
    initial_symbol = np.random.choice([-1, 1]) + 1j * np.random.choice([-1, 1])
    symbols = [initial_symbol]
    for _ in range(1, num_symbols):
        symbol = np.random.choice([-1, 1]) + 1j * np.random.choice([-1, 1])
        symbols.append(np.exp(1j * np.angle(symbol) + 1j * np.pi / 4))
    return np.array(symbols)
```

Функція для модуляції сигналу MSK

```
def modulate_msk_signal(symbols):
    return np.exp(1j * np.pi / 4) * symbols
```

Функція для фільтрації сигналу через фільтр підвищеного косинуса

```
def filter_signal(signal, filter_taps):
    return np.convolve(signal, filter_taps, mode='same')
```

Функція для побудови графіків спектральної щільності потужності (PSD)

```
def plot_psd(signal, title):
    plt.psd(signal, NFFT=1024, Fs=1)
    plt.title(title)
    plt.xlabel('Frequency')
    plt.ylabel('PSD (dB/Hz)')
    plt.grid(True)
    plt.show()
```

Генерація фільтра підвищеного косинуса з коефіцієнтом згортання $\alpha = 0.3$

```
alpha = 0.3
rcos_filter = generate_rcos_filter(alpha)
```

```

# Генерація сигналів та їх модуляція
num_symbols = 1000
qpsk_signal = generate_qpsk_signal(num_symbols)
oqpsk_signal = modulate_oqpsk_signal(qpsk_signal)
p4_dqpsk_signal = generate_p4_dqpsk_signal(num_symbols)
msk_signal = modulate_msk_signal(qpsk_signal)

# Фільтрація сигналів через фільтр підвищеного косинуса
filtered_qpsk_signal = filter_signal(qpsk_signal, rcos_filter)
filtered_oqpsk_signal = filter_signal(oqpsk_signal, rcos_filter)
filtered_p4_dqpsk_signal = filter_signal(p4_dqpsk_signal, rcos_filter)
filtered_msk_signal = filter_signal(msk_signal, rcos_filter)

# Побудова графіків PSD
plot_psd(filtered_qpsk_signal, 'QPSK PSD after RC Filter ( $\alpha = 0.3$ )')
plot_psd(filtered_oqpsk_signal, 'O-QPSK PSD after RC Filter ( $\alpha = 0.3$ )')
plot_psd(filtered_p4_dqpsk_signal, 'p=4-DQPSK PSD after RC Filter ( $\alpha = 0.3$ )')
plot_psd(filtered_msk_signal, 'MSK PSD after RC Filter ( $\alpha = 0.3$ )')

```

У цьому коді для побудови графіків PSD використовується функція `plt.psd`, яка автоматично обчислює PSD для сигналу та побудовує відповідний графік.

Функція `plotWelchPSD` використовується тут для побудови графіків PSD сигналів з модуляцією BPSK, QPSK і MSK з несучою частотою 800 Гц, а отриманий графік показано на рисунку 3.2.

```
def bpsk_qpsk_msk_psd():
    # Usage:
    # >> from chapter_2.psd_estimates import bpsk_qpsk_msk_psd
    # >> bpsk_qpsk_msk_psd()
    from DigiCommPy.passband_modulations import bpsk_mod, qpsk_mod, msk_mod
    from DigiCommPy.essentials import plotWelchPSD
    N=100000 # Number of symbols to transmit
    fc=800;OF =8 # carrier frequency and oversampling factor
    fs = fc*OF # sampling frequency

    a = np.random.randint(2, size=N) # uniform random symbols from 0's and 1's
    (s_bb,t) = bpsk_mod(a,OF) # BPSK modulation(waveform) - baseband
    s_bpsk = s_bb*np.cos(2*np.pi*fc*t/fs) # BPSK with carrier
    s_qpsk = qpsk_mod(a,fc,OF)['s(t)'] # conventional QPSK
    s_msk = msk_mod(a,fc,OF)['s(t)'] # MSK signal

    # Compute and plot PSDs for each of the modulated versions
    fig, ax = plt.subplots(1, 1)
    plotWelchPSD(s_bpsk,fs,fc,ax = ax,color = 'b',label='BPSK')
    plotWelchPSD(s_qpsk,fs,fc,ax = ax,color = 'r',label='QPSK')
    plotWelchPSD(s_msk,fs,fc, ax = ax,color = 'k',label='MSK')
    ax.set_xlabel('$f-f_c$');ax.set_ylabel('PSD (dB/Hz)');ax.legend();fig.show()
```

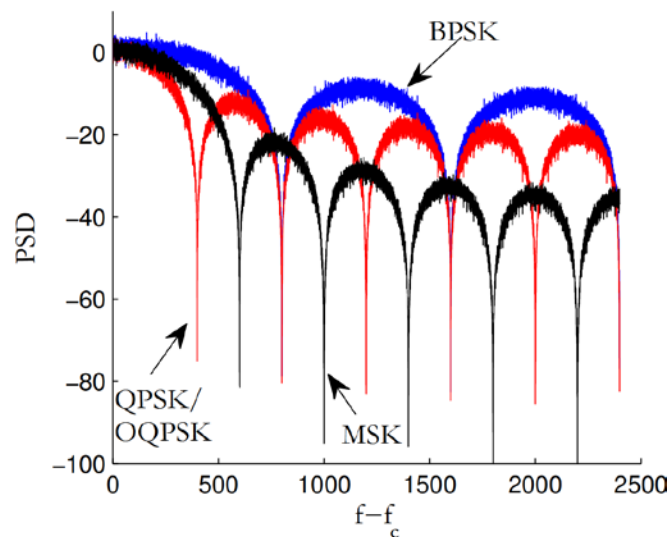


Рисунок 3.2 – Оцінки PSD для сигналів BPSK, QPSK і MSK

Так, мінімальна маніпуляція (MSK) є одним із варіантів фазової маніпуляції з індексом модуляції ($h = 0,5$). Вона використовується в різних системах передачі даних через радіо, зокрема в GSM (Global System for Mobile Communications) і DECT (Digital Enhanced Cordless Telecommunications).

MSK має декілька переваг, таких як постійна огибаюча (constant envelope), компактний спектр та добрі характеристики частоти помилок. Однак основною

проблемою MSK є те, що її спектр не є досить компактним, щоб задовольнити жорсткі вимоги щодо позасмугового випромінювання в таких високошвидкісних системах, як GSM і DECT.

Ці технології мають дуже велику пропускну здатність радіочастотного каналу і вимагають ефективного використання спектра. У таких випадках MSK може бути не досить ефективним через обмежену спектральну ефективність.

Для вирішення цієї проблеми були розроблені інші методи, такі як Gaussian Minimum Shift Keying (GMSK), який використовує гаусівський фільтр для зменшення спектральної ширини сигналу і поліпшення його властивостей в умовах обмеженого спектра.

На графіку спектра MSK (рис. 3.2) видно, що бічні пелюстки зі значною енергією випромінювання виходять далеко за межі швидкості передачі даних. Це проблематично, оскільки викликає сильні позасмугові завади в системах з близько розташованими сусідніми каналами [10].

Премодуляційний фільтр нижніх частот (ФНЧ) може бути ефективним інструментом для маніпулювання спектром сигналу MSK. Гаусівський ФНЧ, зазвичай, задовольняє багатьом вимогам для оптимального управління спектром.

Гаусівський ФНЧ має властивість швидко загасати поза межами заданої смуги частот. Це допомагає позбутися від небажаних високочастотних складових, що дозволяє зберегти спектральну ефективність сигналу. Зменшення перерегулювання в імпульсній характеристиці важливо для захисту від надмірних миттєвих змін частоти, що може призвести до спотворень сигналу. Гаусівський ФНЧ має більш плавний характер, що допомагає уникнути таких проблем. Збереження області вихідного імпульсу фільтра дозволяє застосовувати когерентне детектування, що є бажаним у деяких системах зв'язку, таких як GSM і DECT. Гаусівський ФНЧ володіє оптимальними властивостями для оптимізації спектра сигналу MSK, дозволяючи підтримувати хороші показники спектральної ефективності та виконувати жорсткі вимоги щодо позасмугового випромінювання.

3.3 Висновки до розділу 3

Оцінювання спектральної щільності потужності (PSD) модульованих телекомунікаційних сигналів є важливим етапом у дизайні та аналізі систем передачі даних. PSD надає інформацію про розподіл потужності сигналу в частотній області. Це дозволяє інженерам зв'язку зрозуміти, як сигнал розподіляє свою потужність по різних частотах, що важливо для ефективного використання радіочастотного спектру.

Для оцінювання PSD часто використовуються різні методи, такі як метод Велша, періодограма, метод Бартлетта та інші. Вибір конкретного методу може залежати від характеристик сигналу та специфічних вимог дослідження. Тип та параметри модуляції суттєво впливають на спектральні характеристики сигналу. Наприклад, модуляція з фіксованою амплітудою, така як частотна маніпуляція (FM), може мати інші спектральні характеристики порівняно з модуляцією зі змінною амплітудою, такою як амплітудна маніпуляція (AM).

PSD може бути використаний для керування спектральною ефективністю сигналу, зокрема для зменшення спектрального перекриття з іншими сигналами та використання доступного спектру з більшою ефективністю. Оцінка PSD також дозволяє врахувати вимоги до каналу передачі, такі як обмеження на ширину смуги та вимоги щодо позасмугового випромінювання.

В цілому, оцінка спектральної щільності потужності модульованих телекомунікаційних сигналів є важливим елементом процесу проектування та аналізу систем передачі даних, допомагаючи забезпечити ефективне використання радіочастотного спектру та відповідність вимогам до передавальної системи.

4 КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ЧАСТОТНОЇ МАНІПУЛЯЦІЇ (FSK) РАДІОСИГНАЛІВ

Частотна ключова модуляція (FSK) - це схема модуляції, в якій цифровий сигнал кодується змінами частоти несучого сигналу. У найпростішій конфігурації FSK, відомій як Binary FSK (BFSK), використовуються дві дискретні частоти для представлення двійкових даних - одна частота для 0 і інша для 1. Це може бути дуже ефективним для передачі цифрових даних в бездротових системах. Коли кількість використовуваних частот збільшується до більш ніж двох, метод називається М-арним FSK (MFSK), де "М" позначає кількість різних частот, які використовуються для кодування даних. В MFSK кожна з цих частот може представляти різні символи або комбінації бітів, що дозволяє передавати більше інформації за один символ. FSK застосовується в різноманітних системах зв'язку, включаючи радіо, телефонію, бездротовий Інтернет та інші. Вона має властивості простоти реалізації та стійкості до шуму, що робить її популярним вибором для багатьох застосувань [11].

4.1 Дискретна еквівалентна модель BFSK модуляції

Когерентний і некогерентний дискретний BFSK (Binary Frequency Shift Keying) - це два підходи до демодуляції BFSK-сигналів на приймачі.

В когерентному підході приймач має повну або часткову інформацію про фазу несучого сигналу. На приймачі використовується когерентне демодулювання, що передбачає синхронізацію фази прийнятого сигналу з фазою локального осцилятора. Приймач отримує фазу початкового сигналу від передавача, що дозволяє виконати когерентне демодулювання і отримати правильність передачі даних.

У некогерентному підході приймач не має інформації про фазу несучого сигналу. Демодуляція здійснюється без синхронізації фази, тобто фаза локального осцилятора не синхронізується з фазою прийнятого сигналу. Такий

підхід може бути менш ефективним, але його перевагою є простота реалізації та менші вимоги до системи.

Вибір між когерентним і некогерентним підходом залежить від конкретних вимог до системи зв'язку, складності реалізації та потреб у точності демодуляції. Когерентний підхід, хоча й більш складний, може забезпечувати кращу продуктивність у вимірюваннях умов безперешкодної передачі сигналу.

Дискретно-часова еквівалентна модель для генерування ортогонального BFSK сигналу показана на рисунку 4.1. Тут частота дискретизації (f_s) вибрана досить високою порівняно з несучою частотою (f_c). Кількість дискретних відліків у бітовому періоді L вибирається таким чином, щоб кожен бітовий період був представлений достатньою кількістю відліків, щоб вмістити кілька циклів частот f_1 і f_2 .

Функція `'bfsk_mod'` генерує дискретно-часову еквівалентну модель для генерації Binary FSK (BFSK) сигналу, показану на рисунку 4.1. Вона може підтримувати як когерентну, так і некогерентну форму BFSK-сигналу, в залежності від вказаних параметрів [12].

Якщо вибрано когерентне BFSK, функція повертає згенеровану початкову випадкову фазу ϕ на додаток до згенерованого сигналу BFSK. Ця інформація про фазу вирішальна для когерентного детектування на приймачі, оскільки дозволяє зіставити фазу прийнятого сигналу з фазою початкового сигналу для здійснення оптимального демодулювання.

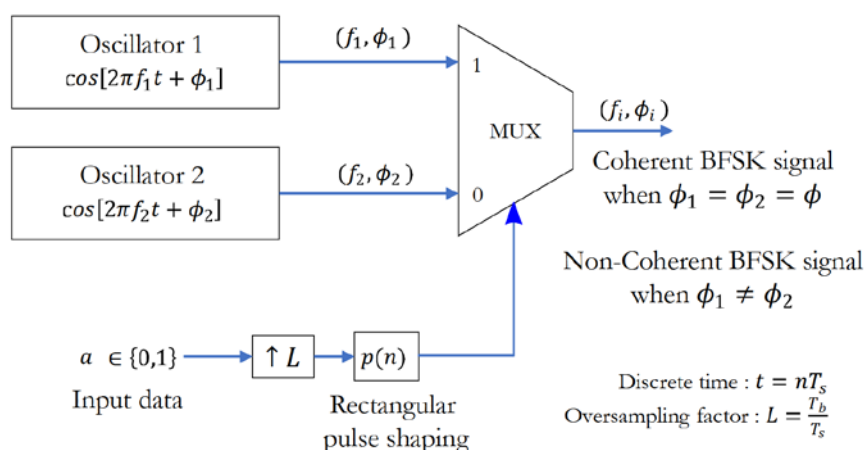


Рисунок 4.1 – Дискретна еквівалентна модель для BFSK модуляції

Отже, функція `bfsk\_mod` надає засоби для моделювання та генерації BFSK-сигналів з різними характеристиками, які можуть бути використані для дослідження та аналізу в системах зв'язку.

```
def bfsk_mod(a,fc,fd,L,fs,fsk_type='coherent',enable_plot = False):
    """
    Function to modulate an incoming binary stream using BFSK
    Parameters:
        a : input binary data stream (0's and 1's) to modulate
        fc : center frequency of the carrier in Hertz
        fd : frequency separation measured from Fc
        L : number of samples in 1-bit period
        fs : Sampling frequency for discrete-time simulation
        fsk_type : 'coherent' (default) or 'noncoherent' FSK generation
        enable_plot: True = plot transmitter waveforms (default False)
    Returns:
        (s_t,phase) : tuple containing following parameters
            s_t : BFSK modulated signal
            phase : initial phase generated by modulator, applicable only for
    → coherent FSK. It can be used when using coherent detection at Rx
    """
    from scipy.signal import upfirdn
    a_t = upfirdn(h=[1]*L, x=a, up = L) #data to waveform

    t = np.arange(start=0,stop=len(a_t))/fs #time base
    if fsk_type.lower() == 'noncoherent':
        # carrier 1 with random phase
        c1 = np.cos(2*np.pi*(fc+fd/2)*t+2*np.pi*np.random.random_sample())
        # carrier 2 with random phase
        c2 = np.cos(2*np.pi*(fc-fd/2)*t+2*np.pi*np.random.random_sample())
    else: #coherent is default
        # random phase from uniform distribution [0,2pi)
        phase=2*np.pi*np.random.random_sample()
        c1 = np.cos(2*np.pi*(fc+fd/2)*t+phase) # carrier 1 with random phase
        c2 = np.cos(2*np.pi*(fc-fd/2)*t+phase) # carrier 2 with the same random phase
    s_t = a_t*c1 +(-a_t+1)*c2 # BFSK signal (MUX selection)

    if enable_plot:
        fig, (ax1,ax2)=plt.subplots(2, 1);ax1.plot(t,a_t);ax2.plot(t,s_t);fig.show()
    return (s_t,phase)
```

Навіть якщо BFSK сигнал був модульований когерентно на передавачі, це не означає, що він має бути когерентно демодульований на приймачі. Некогерентна демодуляція може бути застосована до когерентно модульованих сигналів BFSK на приймачі, особливо у випадках, коли вимоги до точності фазової синхронізації обмежені або коли немає достатньої інформації про фазу сигналу на приймачі.

Фазовий аргумент, що повертається функцією модуляції BFSK на передавачі, дійсно не має значення на приймачі у випадку некогерентної демодуляції, оскільки некогерентна демодуляція ігнорує фазу несучого сигналу.

Такий підхід є зручним у випадках, коли точна фазова синхронізація не є критичною або коли вимоги до складності приймача не дозволяють виконати когерентну демодуляцію. Однак, варто враховувати, що некогерентна демодуляція може мати обмеження щодо продуктивності та якості отриманих результатів порівняно з когерентною демодуляцією [1].

4.2 Когерентний демодулятор

Когерентний демодулятор на основі генератора з керованою напругою (VCO) є одним із найпоширеніших методів демодуляції когерентного сигналу FSK. У цьому методі VCO змінює свою частоту відповідно до вхідного сигналу, і потім вихід VCO демодулюється для відновлення даних. Крім того, когерентний демодулятор може бути реалізований за допомогою корелятора. У цьому випадку корелятор порівнює вхідний сигнал з локальним осцилятором, що має таку ж частоту, як і несуча частота сигналу FSK. Результат кореляції дозволяє визначити, який символ був переданий. Однак важливо враховувати, що для успішної когерентної демодуляції потрібна точна синхронізація фази між передавачем і приймачем. Якщо ця фазова інформація не доступна або не точна, це може призвести до некоректної демодуляції сигналу [2].

Тому, для уникнення проблем з фазовою синхронізацією чи випадкових змін фази несучої хвилі, може бути застосований некогерентний демодулятор. Хоча він може бути менш ефективним, ніж когерентний, він все ж може бути використаний для демодуляції сигналів FSK в умовах, де точна фазова інформація не є обов'язковою або недоступною.

Когерентний демодулятор для FSK також може бути реалізований з використанням лише одного корелятора, і його дискретно-часова еквівалентна модель показана на рисунку 4.2. Тут демодулятору потрібна початкова фазова

інформація ($f_1 = f_2 = f$), яка була використана для генерації когерентного сигналу FSK на передавачі. Щоб уникнути відновлення фази несучої, некогерентний демодулятор може бути використаний для демодуляції когерентного сигналу FSK, але при цьому погіршується продуктивність [3].

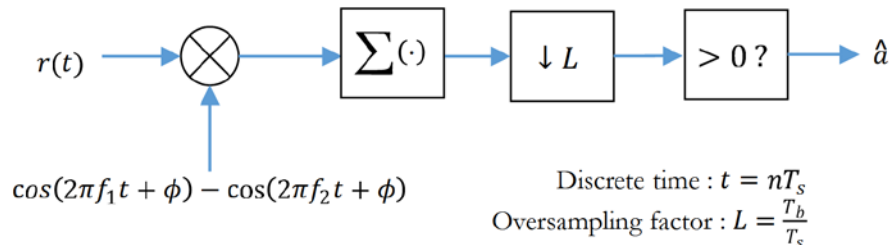


Рисунок 4.2 – Дискретна еквівалентна модель для когерентної демодуляції когерентної FSK

Когерентний демодулятор для когерентної BFSK (Binary Frequency Shift Keying) використовується для демодуляції сигналу BFSK на приймачі з точною синхронізацією фази та частоти несучої хвилі. Наведемо основні кроки, які зазвичай включаються в реалізацію когерентного демодулятора для когерентної BFSK.

Перед початком демодуляції приймач спробує синхронізувати частоту локального осцилятора з частотою несучої хвилі сигналу BFSK. Це може бути досягнуто шляхом використання алгоритмів слідування фази або методів PLL (Phase-Locked Loop). Після синхронізації частоти необхідно забезпечити синхронізацію фази між приймачем і передавачем. Це зазвичай виконується шляхом використання методів відновлення фази або алгоритмів когерентної демодуляції. Після синхронізації приймач використовує отриману інформацію про частоту та фазу несучої хвилі для демодуляції сигналу BFSK. Це може включати в себе порівняння вхідного сигналу зі сигналом локального осцилятора та визначення біта відповідно до результату порівняння. Нарешті, отримані сигнали можуть бути піддані фільтрації та декодуванню для відновлення оригінального потоку даних [4].

В цілому, когерентний демодулятор для когерентної BFSK вимагає точної синхронізації фази та частоти між приймачем і передавачем для ефективної демодуляції сигналу. Це зазвичай забезпечує кращу продуктивність порівняно з некогерентними методами, але вимагає більшої обчислювальної потужності та складнішої реалізації.

```
def bfsk_coherent_demod(r_t, phase, fc, fd, L, fs):
    """
    Coherent demodulation of BFSK modulated signal
    Parameters:
        r_t : BFSK modulated signal at the receiver r(t)
        phase : initial phase generated at the transmitter
        fc : center frequency of the carrier in Hertz
        fd : frequency separation measured from Fc
        L : number of samples in 1-bit period
        fs : Sampling frequency for discrete-time simulation
    Returns:
        a_hat : data bits after demodulation
    """
    t = np.arange(start=0, stop=len(r_t))/fs # time base
    x = r_t*(np.cos(2*np.pi*(fc+fd/2)*t+phase)-np.cos(2*np.pi*(fc-fd/2)*t+phase))
    y = np.convolve(x, np.ones(L)) # integrate/sum from 0 to L
    a_hat = (y[L-1::L]>0).astype(int) # sample at every sampling instant and detect
    return a_hat
```

4.3 Некогерентний демодулятор

Некогерентний демодулятор FSK є простішим у реалізації і ефективним для багатьох практичних застосувань через його здатність працювати без потреби у точній фазовій синхронізації між передавачем і приймачем. Наведемо переваги некогерентного демодулятора FSK.

Некогерентний демодулятор не вимагає точної фазової синхронізації, тому його реалізація є простішою і менш витратною з точки зору обчислювальних ресурсів. Некогерентний демодулятор не обмежений впливом фазових шумів або змінами фази несучої хвилі, що робить його більш стійким до змін у середовищі передачі. Навіть при невеликій якості сигналу або низькому співвідношенні сигнал/шум некогерентний демодулятор може продемонструвати прийнятну продуктивність. Некогерентний демодулятор не потребує знання початкової фазової інформації, що полегшує його використання в різних умовах передачі.

У загальному, некогерентний демодулятор FSK є привабливим вибором для застосувань, де важко або неможливо забезпечити точну фазову синхронізацію, або коли простота реалізації має велике значення. Дискретно-часова еквівалентна модель некогерентного демодулятора на основі корелятора/квадратичного закону показана на рисунку 4.3. Він може бути використаний для демодуляції як когерентного FSK, так і некогерентного FSK сигналу. Для некогерентної демодуляції сигналу FSK можна використовувати дискретну еквівалентну модель на основі квадратичного закону. Основна ідея полягає в тому, що ми використовуємо квадратичний закон детектора для визначення енергії сигналу на двох різних частотах, які використовуються у сигналі FSK. Наведемо алгоритм для побудови такої моделі.

Спочатку отримуємо вхідний сигнал FSK, який може бути відфільтрований та амплітудно нормалізований. Використовуючи квадратичний закон детектора, визначаємо енергію сигналу на кожній з двох можливих частот. Порівнюємо отримані значення енергії. Сигнал з більшою енергією вважається "логічною 1", а меншою - "логічною 0". За отриманими значеннями енергії декодуємо біти, розпізнаючи, який сигнал відповідає "0", а який - "1" [5].

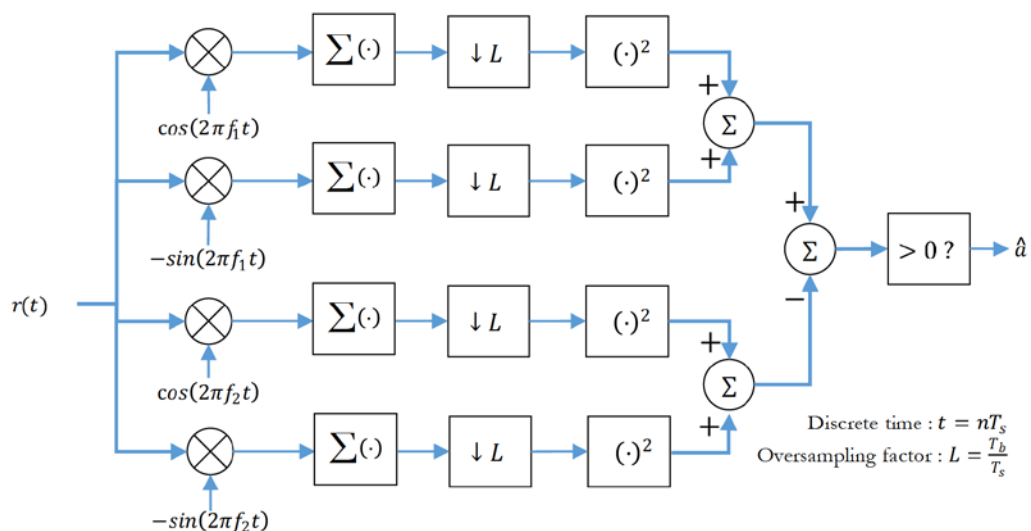


Рисунок 4.3 – Дискретна еквівалентна модель некогерентної демодуляції на основі квадратичного закону для когерентного/некогерентного сигналу FSK

Ця дискретна модель дозволяє демодулювати сигнал FSK навіть без точної фазової синхронізації, що робить її привабливим варіантом для практичних застосувань, де точність синхронізації може бути проблемою. Також вона є менш обчислювально витратною порівняно з когерентними методами, що може бути важливим для реалізації на пристроях з обмеженими ресурсами [6].

Наведемо лістинг програми роботи некогерентного демодулятора на основі квадратичного закону.

```
def bfsk_noncoherent_demod(r_t,fc,fd,L,fs):
    """
    Non-coherent demodulation of BFSK modulated signal
    Parameters:
        r_t : BFSK modulated signal at the receiver r(t)
        fc : center frequency of the carrier in Hertz
        fd : frequency separation measured from Fc
        L : number of samples in 1-bit period
        fs : Sampling frequency for discrete-time simulation
    Returns:
        a_hat : data bits after demodulation
    """
    t = np.arange(start=0,stop=len(r_t))/fs # time base
    f1 = (fc+fd/2); f2 = (fc-fd/2)
    #define four basis functions
    p1c = np.cos(2*np.pi*f1*t); p2c = np.cos(2*np.pi*f2*t)
    p1s = -1*np.sin(2*np.pi*f1*t); p2s = -1*np.sin(2*np.pi*f2*t)
    # multiply and integrate from 0 to L
    r1c = np.convolve(r_t*p1c,np.ones(L)); r2c = np.convolve(r_t*p2c,np.ones(L))
    r1s = np.convolve(r_t*p1s,np.ones(L)); r2s = np.convolve(r_t*p2s,np.ones(L))

    # sample at every sampling instant
    r1c = r1c[L-1::L]; r2c = r2c[L-1::L]
    r1s = r1s[L-1::L]; r2s = r2s[L-1::L]
    # square and add
    x = r1c**2 + r1s**2
    y = r2c**2 + r2s**2
    a_hat=((x-y)>0).astype(int) # compare and decide
    return a_hat
```

4.4 Спектральна густина потужності

Припускаючи прямокутну форму імпульсу в базовій смузі, графік спектральної щільності потужності для BFSK-модульованого сигналу було змодельовано за допомогою віконного методу оцінки спектра Велша і результат

показано на рисунку 4.5. Для цієї симуляції було обрано центральну частоту несучої $f_c = 400$ Гц, частоту дискретизації $f_s = 8 f_c$, індекс модуляції $h = 1$ і бітовий період $T_b = 0.0125$. Як і очікувалося для цієї конфігурації, графік PSD показує два піки на частотах $f_1 = 440$ Гц і $f_2 = 360$ Гц, які відокремлені один від одного на відстань $\Delta f = 80$ Гц [7].

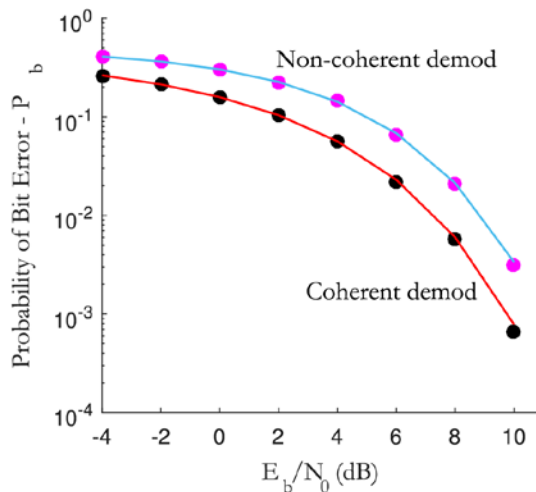


Рисунок 4.4 – Ефективність когерентної BFSK при демодуляції з використанням когерентних і некогерентних методів

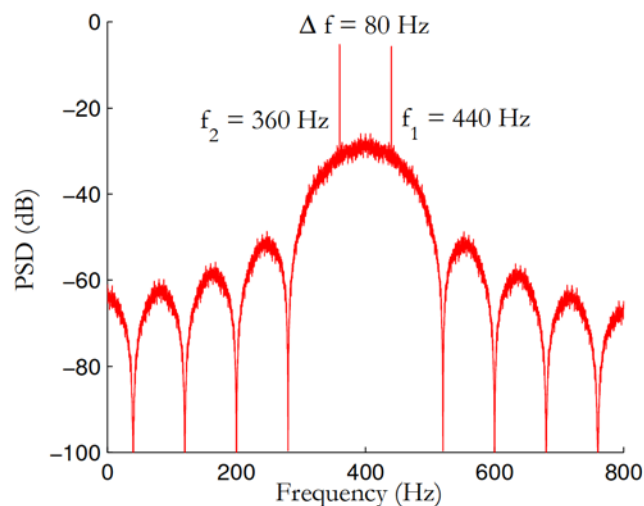


Рисунок 4.5 – Енергетичні спектри BFSK

Енергетичний спектр BFSK (Binary Frequency Shift Keying) визначає, як енергія сигналу розподіляється по частоті. Оскільки BFSK використовує дві різні частоти для представлення двійкових даних, енергетичний спектр буде мати два піки на цих частотах.

Зазвичай, енергетичний спектр BFSK представляється як два дельта-піка, розташованих на частотах, що відповідають двом можливим значенням біта. Відстань між цими піками дорівнює різниці між частотами несучої хвилі, яка використовується для представлення "0" і "1". Ширина піків визначається символною швидкістю передачі (symbol rate) сигналу. Енергетичний спектр BFSK можна виразити у вигляді функції, яка має значення лише на двох позиціях частоти (частоти "0" і "1"), де кожне значення відображає енергію сигналу на відповідній частоті. Така функція зазвичай має вигляд двох дельта-функцій або двох площин-гілок. Енергетичний спектр BFSK є важливим аспектом при проектуванні та аналізі систем зв'язку, оскільки він визначає спектральну ефективність сигналу та його здатність до співіснування з іншими сигналами в каналі [8].

4.5 Висновки до розділу 4

Результати моделювання можуть допомогти оцінити ефективність передачі даних за допомогою FSK у конкретних умовах каналу зв'язку. Це включає аналіз швидкості передачі даних, швидкості помилок передачі та інші параметри продуктивності. Комп'ютерне моделювання дозволяє визначити спектральну щільність потужності сигналів FSK. Це важливо для визначення спектральної ефективності та здатності сигналу до співіснування з іншими сигналами в каналі.

Моделювання дозволяє врахувати вплив різних видів спотворень та шуму на сигнал FSK. Це допомагає визначити, наскільки добре сигнал може бути розпізнаний та демодульований на приймачі в реальних умовах каналу. Моделювання дозволяє проводити експерименти з різними параметрами системи FSK, такими як розмір кроку частоти, кількість використовуваних

частот, тип фільтрів та інші. Це допомагає знайти оптимальні налаштування для конкретних вимог до системи. Моделювання може використовуватися для оцінки продуктивності системи FSK в реальному часі в різних умовах каналу та параметрах системи. Це дозволяє зробити прогнози щодо поведінки системи перед її реальним впровадженням. Отже, комп'ютерне моделювання частотної маніпуляції (FSK) радіосигналів є корисним інструментом для аналізу та оптимізації систем зв'язку на основі цього типу модуляції.

5 ОХОРОНА ПРАЦІ

Проектування програмного застосунку дослідження формату модуляції в безпроводних системах передачі відбувалася в приміщенні, яке обладнане комп'ютеризованими робочими місцями. На дослідника мали вплив такі небезпечні та шкідливі виробничі фактори:

1. Фізичні: підвищена запиленість та загазованість повітря робочої зони; підвищений рівень шуму на робочому місці; підвищена чи понижена вологість повітря; підвищений рівень статичної електрики; підвищений рівень електромагнітного випромінювання; недостатня освітленість робочої зони.

2. Психофізіологічні: розумове перевантаження; перенапруга аналізаторів; статичне перевантаження.

Відповідно до визначених факторів формуємо рішення щодо безпечного виконання роботи.

5.1 Технічні рішення щодо безпечного виконання роботи

5.1.1 Обладнання робочого місця

Організація робочого простору для проектувальника програмного застосунку дослідження формату модуляції в безпроводних системах передачі повинна відповідати ергономічним вимогам, щоб забезпечити комфортні умови праці. Робоче місце розраховане на сидячу роботу. При цьому враховуються загальні принципи ергономіки, а також особливості та характер трудової діяльності. [14].

Площа, виділена для одного робочого місця з ПК, повинна становити не менш 6 м², а об'єм – не менше 20 м³ [15].

Природне освітлення має здійснюватись через світлові прорізи, орієнтовані переважно на північ чи північний схід, і забезпечувати коефіцієнт природної освітленості (КПО) не нижче, ніж 1,5%.

Виробничі приміщення повинні обладнуватись шафами для зберігання документів, стелажми, тумбами тощо, з урахуванням вимог до площі приміщень.

У приміщеннях з ПК слід щоденно робити вологе прибирання.

Приміщення із ПК мають бути оснащені аптечками першої медичної допомоги.

При приміщеннях із ПК мають бути обладнані побутові приміщення для відпочинку під час роботи, кімната психологічного розвантаження. В кімнаті психологічного розвантаження слід передбачити встановлення пристроїв для приготування й роздачі тонізуючих напоїв, а також місця для занять фізичною культурою.

Конструкція робочого місця користувача ПК повинна забезпечувати підтримку оптимальної робочої пози з такими ергономічними характеристиками: стопи ніг – на підлозі або на підставці для ніг; стегна – у горизонтальній площині; передпліччя – вертикальні; лікті – під кутом 70-90° до вертикальної площини; зап'ястя зігнуті під кутом не більше 20° щодо горизонтальної площини, нахил голови – 15-20° щодо вертикальної площини.

Висота робочої поверхні стола для відеотерміналу повинна перебувати в межах 680-800 мм, а ширина – забезпечувати можливість виконання операцій у зоні досяжності моторного поля.

Робочий стіл для ПК повинен мати простір для ніг висотою не менш 600 мм, шириною не менш 500 мм, глибиною на рівні колін не менш 450 мм, на рівні витягнутої ноги - не менш 650 мм.

Робоче сидіння (стілець, крісло) користувача ПК повинен мати наступні основні елементи: сидіння, спинку й стаціонарні або знімні підлокітники.

Екран монітора й клавіатура повинні розташовуватися на оптимальній відстані від очей користувача, але не ближче 600 мм, з урахуванням розміру алфавітно-цифрових знаків і символів.

Клавіатуру варто розміщати на поверхні стола або на спеціальній, регульованій по висоті, робочій поверхні окремо від стола на відстані 100-300 мм

від краю, найближчого до працівника. Кут нахилу клавіатури повинен бути в межах 5-15°.

При організації праці, пов'язаної з використанням ПК, для збереження здоров'я працюючих, запобігання професійним захворюванням і підтримки працездатності передбачаються внутрішньо змінні регламентовані перерви для відпочинку.

Внутрішньозмінні режими праці й відпочинку містять додаткові нетривалі перерви в періоди, що передують появі об'єктивних і суб'єктивних ознак стомлення й зниження працездатності.

Працюючі з ПК підлягають обов'язковим медичним оглядам: попереднім – при влаштуванні на роботу і періодичним – протягом трудової діяльності. Основними критеріями оцінки придатності до роботи з ПК мають бути показники стану органів зору: гострота зору, показники рефракції, акомодації, стану бінокулярного апарату ока тощо. При цьому необхідно враховувати також стан організму в цілому.

Лінія електромережі для живлення ПК, периферійних пристроїв ПК й устаткування для обслуговування, ремонту й налагодження ПК в приміщенні виконана як окрема групова трипровідна мережа, шляхом прокладання фазових, нульових робочих і нульового захисного провідників. Нульовий захисний провідник використовується для заземлення електроприладів.

Металеві труби й гнучкі металеві рукави заземлені. Заземлення відповідає вимогам Правил безпечної експлуатації електроустановок споживачів [16].

Неприпустимим є:

- експлуатація кабелів і проводів з ушкодженими захисними властивостями за час експлуатації ізоляції; залишення під напругою кабелів і проводів з неізованими провідниками;
- застосування саморобних подовжувачів, що не відповідають вимогам ПУЕ до переносних електропроводів;
- застосування для опалення приміщення нестандартного (саморобного) електронагрівального устаткування або ламп накаливання;

- користування ушкодженими розетками, вимикачами й іншими електровиробами, а також лампами, скло яких має сліди затемнення або здуття;
- підвішування світильників безпосередньо на струмоведучих проводах, обгортання електроламп і світильників папером, тканиною й іншими горючими матеріалами, експлуатація їх зі знятими ковпаками (розсіювачами);
- використання електроапаратури й приладів в умовах, що не відповідають вказівкам (рекомендаціям) підприємств-виготовлювачів.

5.2 Технічні рішення з гігієни праці та виробничої санітарії

5.2.1 Мікроклімат

Робота з проектування програмного застосунку дослідження формату модуляції в безпроводних системах передачі за енерговитратами відноситься до категорії 1а [17]. Допустимі параметри мікроклімату для категорії 1а наведені в табл. 5.1 (відповідно до ДСН 3.3.6.042-99 [18]).

Таблиця 5.1 – Параметри мікроклімату

Період року	Допустимі		
	t, °C	W, %	V, м/с
Теплий	22-28	55	0,1-0,2
Холодний	21-25	75	0,1

Для забезпечення оптимального мікроклімату в приміщенні використовується система опалення та вентиляції повітря. Вимірювання показників мікроклімату проводяться на початку, середині і кінці холодного і теплого періодів року, не рідше трьох разів протягом робочої зміни. При коливаннях цих показників, які можуть бути обумовлені технологічними процесами та іншими факторами, вимірювання також проводяться при найменших і найбільших значеннях термічних навантажень на працівників протягом робочої зміни.

5.2.2 Склад повітря робочої зони

В приміщенні, де здійснюється проектування програмного застосунку дослідження формату модуляції в безпроводних системах передачі можливими забруднювачами повітря може бути офісна техніка та пил, який потрапляє ззовні. ГДК шкідливих речовин, які знаходяться в досліджуваному приміщенні, наведені в таблиці 5.2.

Таблиця 5.2 – ГДК шкідливих речовин у повітрі

Назва речовини	ГДК, мг/м ³	Середньо добова	Клас небезпечності
	Максимально разова		
Фенол	0,01	0,01	3
Формальдегід	0,035	0,003	2
Пил нетоксичний	0,5	0,15	4
Озон	0,16	0,03	4

В повітрі зовнішнього природного середовища, як і в повітряному середовищі приміщень завжди є наявною певна кількість заряджених частинок – іонів. Так в 1 см³ чистого зовнішнього повітря міститься близько 1000 негативних іонів і понад 1200 позитивних. Параметри іонного складу повітря на робочому місці, що обладнане ПК, повинні відповідати допустимим нормам (табл.5.3).

Таблиця 5.3 – Рівні іонізації повітря приміщень при роботі на ПК

Рівні	Кількість іонів в 1 см ³	
	n+	n-
Мінімально необхідні	400	600
Оптимальні	1500-3000	3000-5000
Максимально необхідні	50000	50000

Для дотримання нормального складу повітря робочої зони в приміщенні використовують припливно-витяжну вентиляцію. Систематично здійснюють провітрювання через віконні отвори та вологе прибирання з метою підтримки

оптимальних умов для праці. Крім того, в планах передбачено встановлення системи кондиціонування, що допоможе забезпечити стабільні параметри температури та вологості в приміщенні.

5.2.3 Виробниче освітлення

Норми освітленості при штучному освітленні та КПО (для III пояса світлового клімату) при природному та сумісному освітленні для виконання роботи зазначені у таблиці 5.4 (відповідно до ДБН В.2.5-28-2018 [19]):

Таблиця 5.4 - Норми освітленості в приміщенні

Характеристика зорової роботи	Найменший розмір об'єкта розрізнення	Розряд зорової роботи	Підрозряд зорової роботи	Контраст об'єкта розрізнення з фоном	Характеристика фона	Освітленість, лк		КПО, e_n , %			
						Штучне освітлення		Природне освітлення		Сумісне освітлення	
						Комбіноване	Загальне	Верхнє або верхнє	Бокове	Верхнє або верхнє і бокове	Бокове
Дуже високої точності	Від 0,15 до 0,3	II	г	великий	світлий	1000	300	7	2,5	4,2	1,5

Місце праці повинно бути розташоване так, щоб уникнути попадання в очі прямого світла. Щоб уникнути світлових відблисків необхідно використовувати обладнання з матовою поверхнею. Для захисту очей від прямого сонячного світла чи джерел штучного освітлення необхідно застосовувати захисні козирки та жалюзі на вікнах.

Для створення оптимальних умов зорової роботи слід враховувати не лише кількість та якість освітлення, а й кольорове оточення. Так, при світлому пофарбуванні інтер'єру завдяки збільшенню кількості відбитого світла рівень

освітленості підвищується на 20 – 40% (при тій же потужності джерел світла), різкість тіней зменшується, покращується рівномірність освітлення.

5.2.4 Виробничий шум

Допустимі рівні шуму та вібрації на місцях праці осіб, що працюють з ПК, встановлені санітарними нормами ДсанПіН 3.3.2-007-98 [20], витяг з яких подано в таблиці 5.5.

Таблиця 5.5 - Допустимі еквівалентні рівні шуму

Вид професійної діяльності, місце праці	Еквівалентні рівні шуму, дБА.
Програмісти	50
Оператори в залах опрацювання інформації на ПК та оператори комп'ютерного набору	65
В приміщеннях для розташування шумних агрегатів	75

Основними заходами боротьби з шумом є усунення або ослаблення причин шуму в самому його джерелі у процесі роботи, використання звукопоглинаючих матеріалів, раціональне планування виробничих приміщень.

5.2.5 Виробничі випромінювання

Оскільки проектування проводилося за допомогою ПК, то на робочому місці працівника можливий підвищений рівень електромагнітного випромінювання.

Основою функціонування організму є дуже слабкі біоелектричні струми, що синхронізують природні біологічні режими. Штучні ЕМП якщо співпадають з частотами біологічних ритмів мозку або біоелектричною активністю серця чи

інших органів людини можуть призвести до десинхронізації функціональних процесів в організмі.

Механізм біологічної дії на організм людини полягає як у тепловому, так і нетепловому специфічному ефекті, тепла дія ЕМП проявляються у підвищенні температури тіла, а також локальному, вибіркового нагріванні тканин, органів, клітин унаслідок переходу електромагнітної енергії у теплову.

Допустимі значення параметрів неіонізуючих електромагнітних випромінювань від монітору комп'ютера представлені в табл. 5.6.

Таблиця 5.6 – Допустимі значення параметрів неіонізуючих електромагнітних випромінювань

Види поля	Допустимі параметри поля		Допустима поверхнева щільність потоку енергії (інтенсивність потоку енергії), Вт/м ²
	за електричною складовою (E), В/м	за магнітною складовою (H), А/М	
Напруженість електромагнітного поля, 6 кГц...3 МГц	50	5	
3 МГц...30МГц	2	-	
30 МГц...5 ГГц	-	-	10
Електромагнітне поле оптичного діапазону в ультрафіолетовій частині спектру: УФ-С (220...280 нм)			0,001
УФ-В (280...320 нм)			0,01
УФ-А (320. ..400 нм)			10,0
в інфрачервоній частині спектру: 0,76... 10,0 мкм			35,0.. .70,0
Напруженість електричного поля ВДТ			20 вВ/м

Для зменшення впливу ЕМП від ПК на дослідника, необхідно дотримуватися регламентованих режимів роботи та відпочинку.

5.3 Пожежна безпека

В приміщенні, де здійснювалося проектування програмного застосунку дослідження формату модуляції в безпроводних системах передачі використовуються тільки негорючі речовини та матеріали у холодному стані, за ступенем вибухопожежної та пожежної небезпеки приміщення відноситься до категорії «Д».

За вогнестійкістю будинок відноситься до другої категорії [21]. Робоча зона приміщення віднесена до класу вибухонебезпечності В-Па та пожежонебезпечності П-Па, оскільки вибухонебезпечна концентрація пилу і волокон може утворюватися лише внаслідок аварії або несправності.

5.3.1 Технічні рішення системи запобігання пожежі

Система запобігання пожежі включає такі основні напрями:

- запобігання утворенню горючого середовища. Досягається: застосуванням герметичного виробничого устаткування; максимально можливою заміною в технологічних процесах горючих речовин та матеріалів негорючими; обмеженням кількості пожежо- та вибухонебезпечних речовин при використанні та зберіганні, а також правильним їх розміщенням; ізоляцією горючого та вибухонебезпечного середовища; організацією контролю за складом повітря в приміщенні та контролю за станом середовища в апаратах; застосуванням робочої та аварійної вентиляції; відведенням горючого середовища в спеціальні пристрої та безпечні місця; застосуванням в установках з горючими речовинами пристроїв захисту від пошкоджень та аварій; використанням інгібувальних (хімічно активні компоненти, що сприяють припиненню пожежі) та

флегматизаційних (інертні компоненти, що роблять середовище негорючим) добавок та ін.

- запобігання виникненню в горючому середовищі (чи внесенню в нього) джерела запалювання. Досягається: використанням устаткування та пристроїв, при роботі яких не виникає джерел запалювання; використанням електроустаткування, що відповідає за виконанням класу пожежо- та вибухонебезпеки приміщень та зон, груп і категорії вибухонебезпечної суміші; виконанням вимог щодо сумісного зберігання речовин та матеріалів; використанням устаткування, що задовольняє вимоги електростатичної іскробезпеки; улаштуванням блискавкозахисту; організацією автоматичного контролю параметрів, що визначають джерела запалювання; використанням швидкодіючих засобів захисного вимкнення; заземленням устаткування, видовжених металоконструкцій; використанням під час роботи з ЛЗР інструментів, що не допускають іскроутворення; ліквідацією умов для самоспалахування речовин і матеріалів; усуненням контакту з повітрям пірофорних речовин; підтриманням температури нагрівання поверхні устаткування пристроїв, речовин та матеріалів, які можуть контактувати з горючим середовищем нижче гранично допустимої (80 % температури самозаймання).

Можливі причини виникнення пожежі у приміщенні, де проводилося проектування кодерів сигналів звукомовлення на базі цифрової модуляції з передбаченням такі:

- несправна електропроводка (іскріння, перегрів провідників, пересихання електроізоляційних матеріалів);

- залишення без нагляду увімкнутих комп'ютерів, обчислювальної техніки та інших електроприладів.

Для запобігання виникнення пожежі доцільні такі заходи:

- призначення осіб, що відповідальні за пожежну безпеку приміщення;

- систематичне проведення повторних протипожежних інструктажів та занять за програмою пожежно-технічного мінімуму з особами, що відповідальні за пожежну безпеку;
- утримання в справному стані засобів протипожежного захисту.

5.3.2 Технічні рішення системи протипожежного захисту

Протипожежний захист промислових об'єктів забезпечується:

- правильним вибором необхідного ступеня вогнестійкості будівельних конструкцій; правильним об'ємно-планувальним рішенням будівель і споруд; розташуванням приміщень та виробництв з урахуванням вимог пожежної безпеки;
- улаштуванням протипожежних перепон у будівлях, системах вентиляції, опалювальних та кабельних комунікаціях;
- спорудженням протидимного захисту;
- забезпеченням евакуації людей;
- використанням засобів пожежної сигналізації, сповіщення та пожежогасіння;
- організацією пожежної охорони об'єкта;
- засобами, що забезпечують успішне розгортання тактичних дій гасіння пожежі.

Досліджуване приміщення має два переносних (порошкових, водопінних або водяних) вогнегасників з масою заряду вогнегасної речовини 5 кг і більше. Крім того, треба передбачати по одному вуглекислотному вогнегаснику з величиною заряду вогнегасної речовини 3 кг і більше:

- на 20 м² площі підлоги в таких приміщеннях: офісні приміщення з ПК, комори, електрощитові, вентиляційні камери та інші технічні приміщення;
- на 50 м² площі підлоги приміщень архівів, машинних залів, бібліотек, музеїв [22].

ВИСНОВКИ

Результати розробки програмного застосунку для дослідження формату модуляції в безпроводних системах передачі можуть бути досить значними для технічного та наукового співтовариства.

Застосунок дозволяє користувачеві порівнювати різні формати модуляції за такими параметрами, як швидкість передачі даних, спектральна ефективність, стійкість до шуму тощо. Це дозволяє визначити оптимальний формат модуляції для конкретних умов передачі. Застосунок дозволяє вивчати вплив різних умов каналу передачі, таких як шум, спотворення сигналу, затухання тощо, на ефективність роботи різних форматів модуляції. Це допомагає розуміти, як канал впливає на якість зв'язку. Застосунок дозволяє прогнозувати продуктивність бездротових мереж при різних умовах експлуатації, таких як зміни навантаження, розташування антен, зміни параметрів каналу тощо. За допомогою застосунку можна проводити експерименти для пошуку оптимальних параметрів системи, таких як ширина смуги, рівень модуляції, кількість антен тощо, для максимізації продуктивності та надійності мережі.

Результати дослідження, отримані за допомогою застосунку, можуть бути використані для прийняття рішень щодо дизайну та оптимізації бездротових мереж, вибору обладнання та технологій модуляції. Загалом, розробка програмного застосунку для дослідження формату модуляції в безпроводних системах передачі може сприяти вдосконаленню та оптимізації технологій бездротового зв'язку, що має велике значення для розвитку телекомунікаційного галузі.

Дослідження різних форматів модуляції є важливим для розвитку та оптимізації бездротових комунікаційних систем. Вони дозволяють ефективно передавати дані через бездротові канали, забезпечуючи надійність та швидкість передачі.

Аналіз різних методів модуляції, таких як AM, FM, PM, QAM, PSK та CDMA, дозволяє визначити їх переваги та недоліки в різних умовах зв'язку. Це допомагає вибрати оптимальний метод для конкретного застосування.

Оптимізація спектральної щільності потужності дозволяє максимізувати ефективне використання радіочастотного спектру, забезпечуючи ефективну передачу даних при мінімальному впливі на інші системи зв'язку та мінімальній електромагнітній забрудненості. Комп'ютерне моделювання FSK дозволяє дослідити та аналізувати цей метод модуляції бездротових сигналів, виявляючи його ефективність та можливість застосування в різних сценаріях зв'язку.

Охорона праці в бездротових комунікаційних системах включає в себе ідентифікацію та оцінку ризиків, захист від небезпек, навчання та інструктаж працівників, моніторинг та аналіз інцидентів, ергономічне проектування робочих місць та постійне вдосконалення системи охорони праці.

Усі ці аспекти важливі для забезпечення ефективного та безпечного функціонування бездротових комунікаційних систем та здоров'я працівників, які працюють у цих системах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Frenzel, L.E. (2016). Principles of Electronic Communication Systems. New York: McGraw- Hill Education.
2. Quadrature Amplitude Modulation (QAM) (2023). What is it? Electrical4U. <https://www.electrical4u.com/quadrature- amplitude- modulation- qam/> (accessed 14 February 2023).
3. Beard, C., Stallings, W., and Tahiliani, M.P. (2015). Wireless Communication Networks and Systems, Global Edition. Pearson.
4. Salvi, S. and Geetha, V. (2019). From light to Li-F i: research challenges in modulation, MIMO, deployment strategies and handover. International Conference on Data Science and Engineering (ICDSE), Patna, India (26–28 September 2019), pp. 107–119.
5. Васильківський, М. В. Програмні технології в інфокомунікаційних системах. Навчальний посібник для студентів спеціальності 172 «Телекомунікації та радіотехніка» : електронний навчальний посіб-ник комбінованого (локального та мережного) використання [Електронний ресурс] / Васильківський М. В., Бортник Г. Г., Кичак В. М. – Вінниця : ВНТУ, 2023. – 141 с.
6. Основи програмування (Python, Java) : лабораторний практикум / Смотр О., Придатко О., Малець І. – Львів : ЛДУ БЖД, 2019. – 134 с.
7. Програмування мовою Python / О.М. Васильєв. — Тернопіль: Навчальна книга – Богдан, 2019. — 504 с.; іл.
8. Васильківський, М., Коломієць, А., & Грабчак, Н. (2022). Дослідження функціональних параметрів інфокомунікаційних мереж 6G. Вісник Хмельницького національного університету, (6), 46–52. <https://www.doi.org/10.31891/2307-5732-2022-315-6-46-52>
9. Шарадкін Д.М., Субач І.Ю., Микитюк А.В. Інструментальні засоби Python для моделювання та си-стемного аналізу часових рядів при вирішенні задач кіберзахисту інформаційно-комунікаційних си-стем: навч. пос. / Шарадкін Д.М.,

Субач І.Ю., Микитюк А.В.; ІСЗЗІ КПП ім. Ігоря Сікорського. – Київ : КПП ім. Ігоря Сікорського, 2023.- 139 с.

10. Березовський В. Є. Чисельні методи з прикладами реалізації мовою Python / В. Є. Березовський, Л. Є.Ковальов, М. О.Медведєва : навчальний посібник. Умань : ВПЦ «Візаві», 2023. 88 с.

11. Васильківський, М., Прикмета, А., Олійник, А., & Нікітович, Д. (2023). Оптимізація інтелектуальних телекомунікаційних мереж. Вісник Хмельницького національного університету, Технічні науки. – 2023. – № 1. (317). – С. 33–41. doi: 10.31891/2307-5732-2023-317-1-33-41

12. М. Васильківський, О. Городецька, Б. Климчук, і В. Говорун, «Стратегії технологічного розвитку апаратного забезпечення інфокомунікаційних радіомереж», ІТКІ, вип. 56, вип. 1, с. 83–91, Бер 2023.

13. ДСТУ OHSAS 18002:2015. Системи управління гігієною та безпекою праці. Основні принципи виконання вимог OHSAS 18001:2007 (OHSAS 18002:2008, IDT). К. : ГП «УкрНИУЦ», 2016. 21 с

14. ДСТУ 8604:2015 Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги. URL: http://online.budstandart.com/ua/catalog/doc-page?id_doc=71028

15. НПАОП 0.00-7.15-18 Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. URL: http://sop.zp.ua/norm_npaop_0_00-7_15-18_01_ua.php.

16. ДСТУБ В.2.5-82:2016. Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом. К. : ДП «УкрНДНЦ», 2016. 109 с

17. Гігієнічна класифікація праці (за показниками шкідливості і небезпеки факторів виробничого середовища від 12.08.1986 № 4137-86. - [Електронний ресурс] - Режим доступу: <http://zakon4.rada.gov.ua/laws/show/v4137400-86>

18. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень. - [Електронний ресурс] - Режим доступу: <http://mozdocs.kiev.ua/view.php?id=1972>

19. ДБН В.2.5-28-2018 Природне і штучне освітлення - [Електронний ресурс] - Режим доступу: <http://document.ua/prirodne-i-shtuchne-osvitlennja-nor8425.html>

20. Про мінімальні вимоги безпеки при роботі з дисплейним обладнанням: 90/270/ЕЭС. Брюссель : Рада Європейських співтовариств, 1990. URL: <http://docs.pravo.ru/document/view/32704903/>

21. ДБН В.1.1-7:2016 Пожежна безпека об'єктів будівництва. Загальні вимоги. URL: http://www.poliplast.ua/doc/dbn_v.1.1-7-2002.pd

22. Наказ Міністерства внутрішніх справ України «Про затвердження Правил експлуатації та типових норм належності вогнегасників». URL: <https://zakon.rada.gov.ua/laws/show/z0225-18#Text>.

ДОДАТКИ

Додаток А
(обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА
ПРОГРАМНИЙ ЗАСТОСУНОК ДОСЛІДЖЕННЯ ФОРМАТУ МОДУЛЯЦІЇ В
БЕЗПРОВІДНИХ СИСТЕМАХ ПЕРЕДАЧІ
назва бакалаврської дипломної роботи

```
def bpsk_qpsk_msk_psd():
    # Usage:
    # >> from chapter_2.psd_estimates import bpsk_qpsk_msk_psd
    # >> bpsk_qpsk_msk_psd()
    from DigiCommPy.passband_modulations import bpsk_mod, qpsk_mod, msk_mod
    from DigiCommPy.essentials import plotWelchPSD
    N=100000 # Number of symbols to transmit
    fc=800;OF=8 # carrier frequency and oversampling factor
    fs = fc*OF # sampling frequency

    a = np.random.randint(2, size=N) # uniform random symbols from 0's and 1's
    (s_bb,t) = bpsk_mod(a,OF) # BPSK modulation(waveform) - baseband
    s_bpsk = s_bb*np.cos(2*np.pi*fc*t/fs) # BPSK with carrier
    s_qpsk = qpsk_mod(a,fc,OF)['s(t)'] # conventional QPSK
    s_msk = msk_mod(a,fc,OF)['s(t)'] # MSK signal

    # Compute and plot PSDs for each of the modulated versions
    fig, ax = plt.subplots(1, 1)
    plotWelchPSD(s_bpsk,fs,fc,ax = ax,color = 'b',label='BPSK')
    plotWelchPSD(s_qpsk,fs,fc,ax = ax,color = 'r',label='QPSK')
    plotWelchPSD(s_msk,fs,fc, ax = ax,color = 'k',label='MSK')
    ax.set_xlabel('$f-f_c$');ax.set_ylabel('PSD (dB/Hz)');ax.legend();fig.show()
```

Рисунок 1 – Лістинг програми оцінювання PSD модульованих сигналів

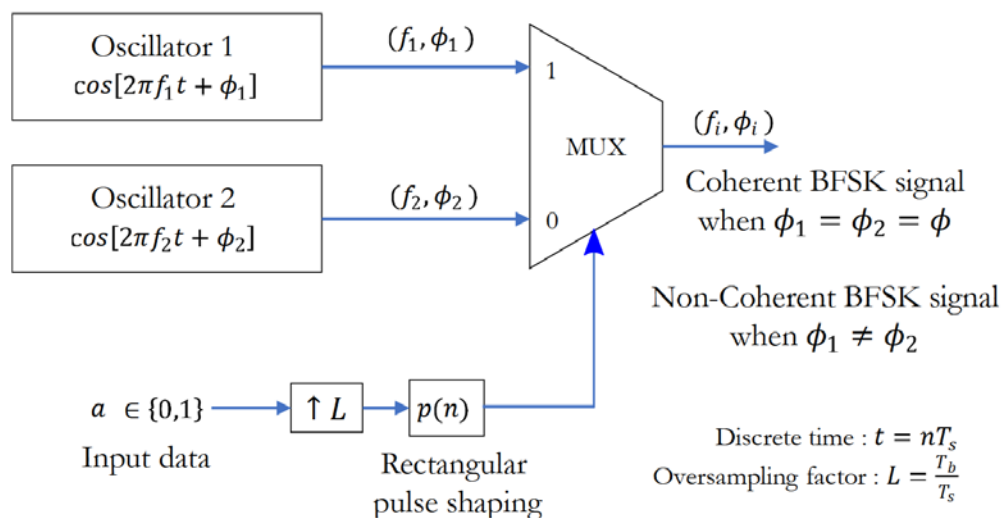


Рисунок 2 – Дискретна еквівалентна модель для BFSK модуляції

```

def bfsk_mod(a,fc,fd,L,fs,fsk_type='coherent',enable_plot = False):
    """
    Function to modulate an incoming binary stream using BFSK
    Parameters:
        a : input binary data stream (0's and 1's) to modulate
        fc : center frequency of the carrier in Hertz
        fd : frequency separation measured from Fc
        L : number of samples in 1-bit period
        fs : Sampling frequency for discrete-time simulation
        fsk_type : 'coherent' (default) or 'noncoherent' FSK generation
        enable_plot: True = plot transmitter waveforms (default False)
    Returns:
        (s_t,phase) : tuple containing following parameters
            s_t : BFSK modulated signal
            phase : initial phase generated by modulator, applicable only for
    → coherent FSK. It can be used when using coherent detection at Rx
    """

    from scipy.signal import upfirdn
    a_t = upfirdn(h=[1]*L, x=a, up = L) #data to waveform

    t = np.arange(start=0,stop=len(a_t))/fs #time base
    if fsk_type.lower() == 'noncoherent':
        # carrier 1 with random phase
        c1 = np.cos(2*np.pi*(fc+fd/2)*t+2*np.pi*np.random.random_sample())
        # carrier 2 with random phase
        c2 = np.cos(2*np.pi*(fc-fd/2)*t+2*np.pi*np.random.random_sample())
    else: #coherent is default
        # random phase from uniform distribution [0,2pi)
        phase=2*np.pi*np.random.random_sample()
        c1 = np.cos(2*np.pi*(fc+fd/2)*t+phase) # carrier 1 with random phase
        c2 = np.cos(2*np.pi*(fc-fd/2)*t+phase) # carrier 2 with the same random phase
    s_t = a_t*c1 +(-a_t+1)*c2 # BFSK signal (MUX selection)

    if enable_plot:
        fig, (ax1,ax2)=plt.subplots(2, 1);ax1.plot(t,a_t);ax2.plot(t,s_t);fig.show()
    return (s_t,phase)

```

Рисунок 3 – Лістинг програми моделювання BFSK телекомунікаційних сигналів

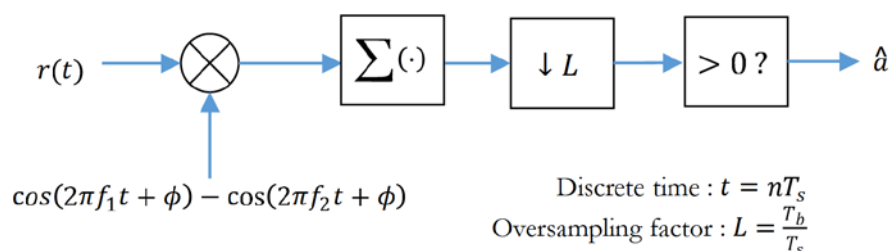


Рисунок 4 - Дискретна еквівалентна модель для когерентної демодуляції когерентної FSK

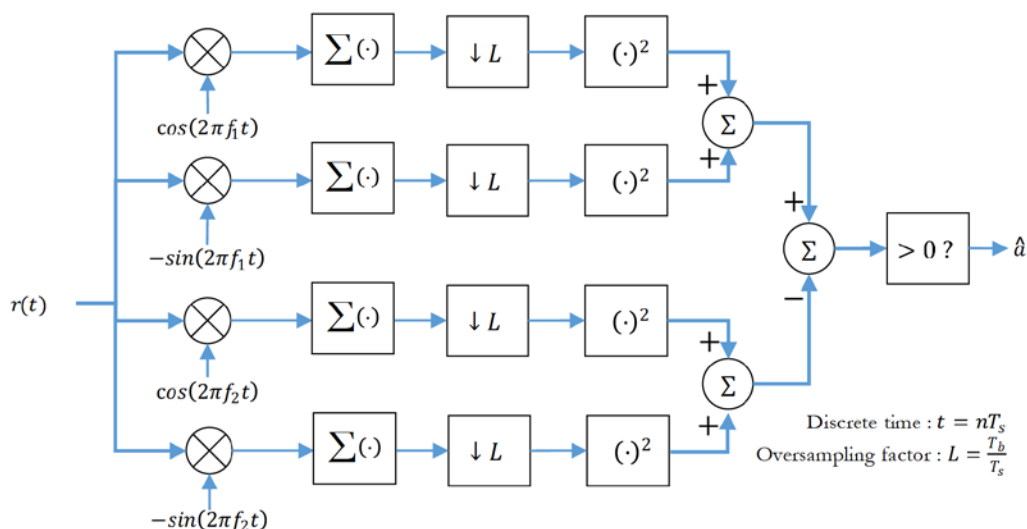


Рисунок 5 - Дискретна еквівалентна модель некогерентної демодуляції на основі квадратичного закону для когерентного/некогерентного сигналу FSK

```
def bfsk_noncoherent_demod(r_t,fc,fd,L,fs):
    """
    Non-coherent demodulation of BFSK modulated signal
    Parameters:
        r_t : BFSK modulated signal at the receiver r(t)
        fc : center frequency of the carrier in Hertz
        fd : frequency separation measured from Fc
        L : number of samples in 1-bit period
        fs : Sampling frequency for discrete-time simulation
    Returns:
        a_hat : data bits after demodulation
    """
    t = np.arange(start=0,stop=len(r_t))/fs # time base
    f1 = (fc+fd/2); f2 = (fc-fd/2)
    #define four basis functions
    p1c = np.cos(2*np.pi*f1*t); p2c = np.cos(2*np.pi*f2*t)
    p1s = -1*np.sin(2*np.pi*f1*t); p2s = -1*np.sin(2*np.pi*f2*t)
    # multiply and integrate from 0 to L
    r1c = np.convolve(r_t*p1c,np.ones(L)); r2c = np.convolve(r_t*p2c,np.ones(L))
    r1s = np.convolve(r_t*p1s,np.ones(L)); r2s = np.convolve(r_t*p2s,np.ones(L))

    # sample at every sampling instant
    r1c = r1c[L-1::L]; r2c = r2c[L-1::L]
    r1s = r1s[L-1::L]; r2s = r2s[L-1::L]
    # square and add
    x = r1c**2 + r1s**2
    y = r2c**2 + r2s**2
    a_hat=((x-y)>0).astype(int) # compare and decide
    return a_hat
```

Рисунок 6 - Лістинг програми роботи некогерентного демодулятора на основі квадратичного закону

Додаток Б
(обов'язковий)

Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень

ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Програмний застосунок дослідження формату модуляції в безпроводних системах передачі

Тип роботи: Бакалаврська дипломна робота
(БДР, МКР)

Підрозділ кафедра інфокомунікаційних систем і технологій, факультет інформаційних електронних систем
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 96,25 % Схожість 3,75 %

Аналіз звіту подібності (відмітити потрібне):

☒ 1. Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.

☐ 2. Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.

☐ 3. Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа відповідальна за перевірку

(підпис)

Васильківський М.В.

(прізвище, ініціали)

Ознайомлені з повним звітом, який був згенерований системою Unicheck щодо роботи.

Автор роботи

(підпис)

Мізяківський В.М.

(прізвище, ініціали)

Керівник роботи

(підпис)

Бортник С.Г.

(прізвище, ініціали)