

Міністерство освіти і науки України  
Національний університет «Запорізька політехніка»  
Кафедра радіотехніки та телекомунікацій

## **МЕТОДИЧНІ ВКАЗІВКИ**

До курсової роботи з дисципліни  
«Широкосмугові технології телекомунікацій»  
Для студентів спеціальності 172 «Телекомунікації та радіотехніка»  
всіх форм навчання

Частина I

2019

Методичні вказівки до курсової роботи з дисципліни «Широкопasmові технології телекомунікацій» для студентів спеціальності 172 «Телекомунікації та радіотехніка» всіх форм навчання. Частина I / Укл. В.С. Кабак, Г.М. Сидоренко – Запоріжжя: НУ «Запорізька політехніка», 2019, – 68 с.

Укладачі: В.С. Кабак, доцент, к.т.н.  
Г.М. Сидоренко, зав. лаб.

Рецензент: С.В. Морщавка, доцент, к.т.н.

Відповідальний.  
за випуск: Г.М. Сидоренко, зав. лаб.

Затверджено:  
на засіданні кафедри  
радіотехніки та телекомунікацій  
Протокол № 1 від 11.09.2019 р.

Рекомендовано до видання НМК  
факультету радіоелектроніки та  
телекомунікацій  
Протокол № 2 від 19.09.2019 р.

**ЗМІСТ**

ПЕРЕЛІК СКОРОЧЕНЬ.....	4
1 ЗМІСТ КУРСОВОЇ РОБОТИ.....	6
1.1 Загальні вимоги до змісту.....	6
1.2 Вимоги до структурних елементів ПЗ.....	7
2 КАНАЛ ЗВ'ЯЗКУ З ВИКОРИСТАННЯМ M-QAM.....	8
2.1 Типи модуляції в цифрових каналах.....	8
2.2 Багатопозиційні методи модуляції.....	14
2.2.1 Метод маніпуляції 4-PSK (QPSK).....	14
2.2.2 Квадратурна амплітудна маніпуляція.....	16
3 АЛГОРИТМ РОБОТИ СИГНАЛЬНОГО КОДЕРА.....	28
4 МОДЕЛЬ КАНАЛУ ЗВ'ЯЗКУ З ВИКОРИСТАННЯМ M-QAM.....	35
4.1 Структурна схема моделі каналу зв'язку з використанням методу M-QAM.....	35
4.2 Реалізація програмного алгоритму для AWGN каналу з QAM модуляцією.....	39
4.3 Опис головної програми.....	41
4.4 Алгоритм розрахунку бітової помилки.....	50
5 ПОРЯДОК ВИКОНАННЯ КУРСОВОЇ РОБОТИ.....	54
5.1 Обґрунтування алгоритму роботи сигнального кодера.....	54
5.2 Експериментальна частина.....	57
ПЕРЕЛІК ПОСИЛАНЬ.....	60
ДОДАТОК А.....	61
ДОДАТОК Б.....	67

## ПЕРЕЛІК СКОРОЧЕНЬ

БПН	–	без повернення до нуля
ЕОМ	–	електронна обчислювальна машина
М-АЧМ	–	М-ічна амплітудно-частотна маніпуляція
М-АФМ	–	М-ічна амплітудно-фазова маніпуляція
М-АФЧМ	–	М-ічна амплітудно-фазочастотна маніпуляція
ПЗ	–	пояснювальна записка
ТЗ	–	технічне завдання
ФМ	–	фазова маніпуляція
ЧМ	–	частотна маніпуляція
ASK	–	(Amplitude Shift Keying) амплітудна маніпуляція
BASK	–	(Binary Amplitude Shift Keying) двійкова амплітудна маніпуляція
BER	–	(Bit Error Rate) коефіцієнт бітової помилки
BFSK	–	(Binary Frequency Shift Keying) двійкова частотна маніпуляція
BPSK	–	(Binary Phase Shift Keying) двійкова фазова маніпуляція
CDMA	–	(Code Division Multiple Access) множинний доступ з кодовим розділенням каналів
DSSS	–	(Direct Sequence Spread Spectrum) широкосмугова модуляція з прямим розширенням спектра
FHSS	–	(Frequency Hopping Spread Spectrum) розширення спектра з швидкою псевдовипадковою перебудовою частоти
FSK	–	(Frequency Shift Keying) – частотна маніпуляція
GMSK	–	(Gaussian Minimum Shift Keying) гаусівська маніпуляція з мінімальним зсувом
PSK	–	(Phase Shift Keying) фазова маніпуляція
M-QAM	–	(M-level Quadrature Amplitude Modulation) М-рівнева квадратурна амплітудна модуляція
NRZ	–	(Non Return to Zero) –без повернення до нуля
OFDM	–	(Orthogonal Frequency Division Multiplexing) мультиплексування ортогонально розділених несучих
PSK	–	(Phase Shift Keying) фазова маніпуляція

- QAM – (Quadrature Amplitude Modulation) квадратурна амплітудна модуляція
- QPSK – (Quadrature Phase Shift Keying) квадратурна фазова маніпуляція

# 1 ЗМІСТ КУРСОВОЇ РОБОТИ

## 1.1 Загальні вимоги до змісту

Курсова робота складається з розрахунково-теоретичної та експериментальної частин. Розрахунково-теоретична частина оформлюється у вигляді пояснювальної записки (ПЗ).

Типовий обсяг записки – 20...30 машинописного тексту, що включає ілюстрації, але можуть бути деякі зміни як у бік зменшення, так і у бік збільшення обсягу.

Експериментальна частина містить результати моделювання за розробленою моделлю каналу зв'язку і лістинги відповідних програм.

Пояснювальну записку умовно поділяють на:

### а) вступну частину:

- 1) титульний лист;
- 2) завдання на курсовий проект;
- 3) реферат;
- 4) зміст;
- 5) перелік умовних позначень, символів, одиниць, скорочень, термінів;

### б) основну частину:

- 1) вступ;
- 2) *суть ПЗ*:
  - теоретичний аналіз;
  - розробка алгоритмів;
  - розрахунок на ЕОМ;
- 3) висновки;
- 4) рекомендації;
- 5) перелік посилань;

### в) додатки.

## 1.2 Вимоги до структурних елементів ПЗ

**Завдання на курсову роботу** оформлюється на друкарському бланку, виконаному на двосторонньому аркуші, що розміщується після титульного листа. У технічному завданні наводяться всі основні вихідні та якісні показники системи, що проектується.

**Реферат** має бути розміщеним безпосередньо за технічним завданням і має містити такі дані:

- об'єкт розробки;
- мета та результати роботи;
- основні технічні характеристики;
- висновки;
- перелік ключових слів;

Ключові слова (від 5 до 15 слів або словосполучень, надрукованих великими літерами в рядок через коми), що є визначальними для розкриття суті ПЗ, розміщують після тексту реферату. Номер сторінки на рефераті не проставляють.

**Зміст** розташовують безпосередньо після реферату, починаючи з нової сторінки. До змісту включають: вступ; послідовно перелічені назви усіх розділів, підрозділів, пунктів і підпунктів (якщо вони мають заголовки), висновки, перелік посилань та назви додатків.

**Перелік умовних позначень** вміщують безпосередньо після змісту, починаючи з нової сторінки. У ньому наводяться малопоширені умовні позначення, символи, одиниці, скорочення і терміни в алфавітному порядку.

**Висновки** розміщують після викладення суті ПЗ, починаючи з нової сторінки. У висновках наводять оцінку одержаних результатів роботи.

**Перелік посилань** наводять у кінці тексту ПЗ, починаючи з нової сторінки. Бібліографічні описи в переліку посилань подають у порядку за яким вони вперше згадуються в тексті.

**Додатки** містять матеріал, який є необхідним для повноти ПЗ, але не може бути розміщеним в основній частині ПЗ через великий обсяг або способи відтворення. У додатки можуть бути включені ілюстрації або таблиці, результати розрахунків на ЕОМ

## 2 КАНАЛ ЗВ'ЯЗКУ З ВИКОРИСТАННЯМ M-QAM

### 2.1 Типи модуляції в цифрових каналах

У лініях зв'язку переносниками інформації є не власне кодовані повідомлення, а модульовані ними допоміжні сигнали. Модуляція – це нелінійне перетворення характеристик і параметрів сигналу-переносника інформації сигналом повідомлення для передавання інформації по лініям зв'язку. Як переносники інформації найчастіше використовують електричні або оптичні сигнали.

Модуляція у системах зв'язку виконує декілька функцій [1].

По-перше, модуляція дозволяє одночасно передавати по одному каналу багато інформаційних повідомлень, які займають приблизно однакову частотну смугу, перенесенням спектра повідомлень у різні області частотного діапазону.

По-друге, модуляція дає змогу реалізувати інформаційний обмін на тих ділянках частотного діапазону, які найсприятливіші для генерації та поширення сигналу по каналу зв'язку.

По-третє, спектри модульованих коливань складніші, займають ширшу смугу частот, ніж спектри власне повідомлень, що підвищує завадостійкість систем передачі інформації.

Зворотну операцію – виділення інформаційного повідомлення з модульованого сигналу – називають демодуляцією або детектуванням. Операції модуляції і демодуляції виконують модулятори і демодулятори відповідно.

Оскільки модуляція (демодуляція) належить до нелінійних перетворень, у складі модуляторів мають бути або нелінійні елементи, або лінійні схеми зі змінними у часі параметрами.

У разі передавання дискретних повідомлень модуляцію використовують для накладання на сигнал-переносник попередньо закодованого повідомлення, при цьому інформаційний параметр несучого коливання (амплітуда, частота або фаза) змінюється стрибкоподібно і протягом тривалості імпульсу первинного сигналу залишається постійним. Такий процес називають маніпуляцією або дискретною модуляцією.

Вплив модуляції на систему полягає не лише у перенесенні спектра інформаційного сигналу у виділений для каналу зв'язку

частотний діапазон, а є набагато ширшим. В іншому випадку усі системи могли б використовувати один вид модуляції і проблема підвищення ефективності системи шляхом вибору виду модуляції та режимів її функціонування відпала би сама собою [1].

Реально ситуація є зовсім протилежною, тому використовуються десятки видів цифрової модуляції. Більше того вплив процесів модуляції/демодуляції на характеристики системи зв'язку є визначальним [2, 3].

В процесі здійснення модуляції наявні дві складові – вхідна інформація та несуче коливання, на базі яких формується модульований сигнал. Розглянемо детальніше ці складові.

Взагалі цифрові системи передачі характеризуються двома важливими особливостями [4,5]:

– по-перше, будь-які повідомлення надаються у цифровій формі, тобто у вигляді послідовності біт  $\{a_i, i = \dots -1, 0, +1 \dots\}$ ; для довільного значення індексу «і» символ  $a_i$  може приймати тільки значення із алфавіту  $\{0,1\}$ ;

– по-друге, передавач системи зв'язку формує і передає по черзі у канал передачі кінцеву кількість сигналів  $\{s_m(t), m = 1, 2, \dots, M\}$ , що відрізняються за формою, які прийнято називати канальними символами; один канальний символ переносить один або більшу кількість інформаційних біт, які належать передачі; якщо  $M = 2$ , то систему передачі називають двійковою, а для  $M > 2$  маємо  $M$ -ічну систему [4].

Тобто, вхідна інформація для цифрової системи передавання представляє собою послідовність бітів (нулів та одиниць), причому тривалість кожного біта становить  $T_b$ . В результаті наявний потік вхідної інформації з швидкістю  $R = 1/T_b$ .

Але, як відзначалося, вхідна інформація не є електричним сигналом, тому вона не може бути представлена у вигляді часової залежності амплітуди напруги або струму і не може здійснювати модуляцію коливання. Вхідну інформацію спочатку необхідно перетворити в електричний сигнал, тобто здійснити кодування вхідної інформації.

Відповідно, в результаті кодування (до формування канальних символів) інформаційні біти перетворюються на електричні імпульси прямокутної, трикутної форм, у вигляді відрізків гармонічного коливання і т. ін. Досить часто для кодування використовується

послідовність додатних і від'ємних імпульсів прямокутної форми без постійної складової (цей сигнальний формат для двійкових сигналів відомий як формат без повернення до нуля – БПН, або англійською NRZ). Отриманий кодовий сигнал в подальшому може використовуватися вже як сигнал, що безпосередньо здійснює модуляцію несучого коливання.

Несуче коливання, як правило, є гармонічним коливанням. Модулюючий сигнал може діяти як окремо на амплітуду, частоту та фазу несучого коливання так і одночасно на декілька параметрів, зокрема, на амплітуду і фазу. В результаті отримуємо амплітудну маніпуляцію (ASK), частотну маніпуляцію (FSK), фазову маніпуляцію (PSK) і комбіновану амплітудно – фазову (QAM) маніпуляцію.

Якщо кожен біт вхідної інформації змінює параметри несучого коливання (тобто, один радіосимвол переносить один інформаційний біт), то отримуємо двійкові або бінарні види модуляції (BASK, BFSK, BPSK).

Але, крім бінарних модуляцій, як було відзначено, використовуються також так звані М-ічні або М-рівневі модуляції, де  $M > 2$ . При цьому декілька вхідних бітів об'єднуються у символи.

Узагальнену класифікацію основних видів маніпульованих сигналів наведено на рис.2.1

По-перше, відзначимо, що на рис.2.1 відокремлені також широкосмугові види модуляції, які використовуються в системах з розширенням спектра – зокрема, розширення спектра методом прямої послідовності DSSS, яке застосовується в системах з використанням множинного доступу з кодовим розділенням (CDMA), розширення спектра методом стрибків по частоті FHSS, яке застосовується, наприклад, в стандарті GSM і у стандарті IEEE 802.15.1 (персональні безпроводні мережі на основі технології Bluetooth), а також системи з використанням розділення на підставі мультиплексування за допомогою ортогональних піднесучих (OFDM), яке знайшло широке застосування у цифровому телебаченні та мережах безпроводного зв'язку (стандарти IEEE 802.11a, g, IEEE 802.16.).

Основною задачею курсового проектування є моделювання каналу зв'язку з використанням різних методів модуляції, як широкосмугових, так і вузькосмугових. Зокрема, в першій частині методичних вказівок розглядаються ефективні багатопозиційні методи модуляції М-QAM, які відносяться до вузькосмугових методів.

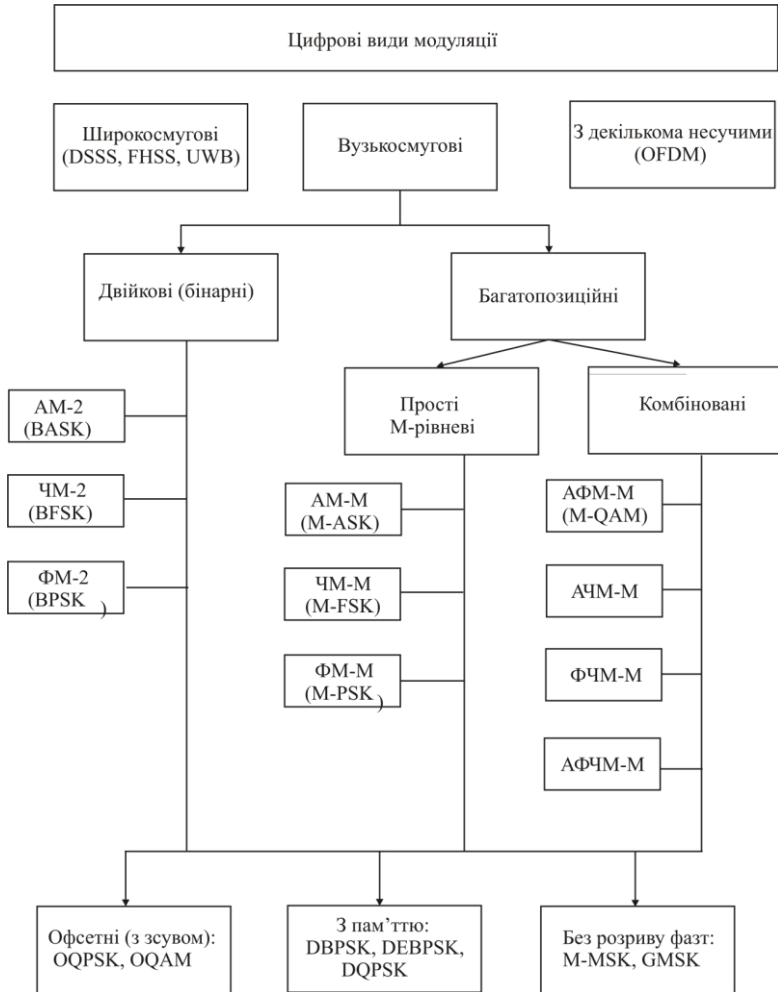


Рисунок 2.1 – Класифікація маніпульованих сигналів

У запропонованій класифікації виділені вже згадані методи двійкової або бінарної маніпуляції і відокремлена ціла низка методів багатопозиційної маніпуляції, які розділені на дві групи – прості і комбіновані методи.

Як відзначалося, для  $M > 2$ , проводиться об'єднання  $k$ -бітів, що в результаті надає  $M=2^k$  різних символів і саме такі види маніпуляції називають багатопозиційними або  $M$ -ічними ( $M$ -рівневими) видами маніпуляції.

В свою чергу багатопозиційні види маніпуляції за кількістю інформативних параметрів сигналу можна розділити на дві основні групи: прості й комбіновані. До простих належать види маніпуляції з одним інформативним параметром –  $M$ -ічна амплітудна маніпуляція  $M$ -АМ (MASK),  $M$ -ічна частотна маніпуляція  $M$ -ЧМ (MFSK),  $M$ -ічна фазова маніпуляція  $M$ -ФМ (MPSK).

До комбінованих у загальному випадку відносять види маніпуляції з кількома інформаційними параметрами –  $M$ -ічна амплітудно-фазова маніпуляція ( $M$ -АФМ, або  $M$ -QAM),  $M$ -ічна амплітудно-частотна маніпуляція ( $M$ -АЧМ),  $M$ -ічна фазочастотна маніпуляція ( $M$ -ФЧМ) і амплітудно-фазочастотна маніпуляція  $M$ -АФЧМ. Основні характеристики модуляції можна розділити на три групи (рис.2.2) [1].



Рисунок 2.2 – Основні характеристики методів модуляції

Якщо часові характеристики (епюри) носять в основному пізнавальний характер, то характеристики енергетичної та частотної ефективності модуляції суттєво впливають на показники системи в цілому.

Відомо, що максимальне значення частотної ефективності системи передавання відеосигналу  $(R/B)_{\max}=2$ , де  $B$ -смуга пропускання каналу [6]. Але при застосуванні модуляції даний показник в більшості випадків значно погіршується. Справа у тому, що

модульований сигнал  $S_m$  представляє собою результат перемноження двох складових – модуляційного сигналу та несучого коливання [7]:

$$S_M = S_m S_c = S_m A_c \cos(\omega_c t + \varphi_c),$$

де  $S_m$  – модулюючий сигнал;

$A_c$ ,  $\omega_c$ ,  $\varphi_c$  – амплітуда, частота та фаза несучого коливання, відповідно.

Відомо [6, 7], що у випадку застосування модулюючого сигналу у вигляді полярного NRZ сигналу ширина смуги частот модульованого сигналу становить  $2f_m$  (де  $f_m$  – верхня частота модулюючого сигналу, який є відеосигналом). В результаті для модульованого сигналу частотна ефективність зменшується порівняно з частотною ефективністю модулюючого сигналу (відеосигналу):

$$(E_f)_m = \left( \frac{R}{B_m} \right)_{\max} = 1. \quad (2.1)$$

Саме на значення частотної ефективності, що визначається виразом (2.1) необхідно орієнтуватися при застосуванні модуляції, крім спеціально оговорених випадків.

Розглянемо частотну ефективність системи при використанні  $M$ -рівневої модуляції. В модуляторі в процесі об'єднання  $k$ -бітів в один символ формуються  $M$ -рівневі види модуляції ( $M = 2^k$ ), причому в даному випадку тривалість символу збільшується, а швидкість передавання відповідно зменшується:

$$T_s = kT_b, \quad (2.2)$$

$$R_s = R/k. \quad (2.3)$$

Як видно із залежності (2.3), при використанні  $M$ -рівневої модуляції ширину смуги частот згідно з критерієм Найквіста можна також зменшити в  $k$ -разів. Але, з даної залежності можна зробити інший, важливий для практики висновок: якщо ширину смуги частот в обох випадках (бінарної та  $M$ -рівневої) залишити однаковою

( $B = \text{const}$ ), то швидкість передавання інформації при використанні  $M$ -рівневих модуляцій збільшується у  $k$ -разів.

Відповідно, при використанні  $M$ -рівневих модуляцій представляють інтерес наступні показники: кількість бітів в символах, ширина смуги частот кожного каналу, швидкість передавання інформації.

## 2.2 Багатопозиційні методи модуляції

Багатопозиційні прості методи маніпуляції передбачають виконання маніпуляції лише для одного інформативного параметра сигналу – амплітуди, частоти або фази. У процесі маніпуляції ці параметри можуть набувати одне з  $M > 2$  дозволених значень.

Відповідно до цього розрізняють  $M$ -ічні амплітудно-маніпульовані, частото-маніпульовані і фазоманіпульовані сигнали. У відомих цифрових системах передачі використовують маніпульовані сигнали з  $M = 4, 8, 16, 32, 64$  і навіть  $256$ .

Це дає змогу значно підвищити швидкість передавання інформації, оскільки кожним символом передається  $\log_2 M$  бітів. Однак такі системи стають більш критичними до дії шумів і завад, оскільки з підвищенням  $M$  зростають вимоги до розрізнення двох сигналів з близькими значеннями інформативних параметрів.

### 2.2.1 Метод маніпуляції 4-PSK (QPSK)

Зупинимося детальніше на відносно простій завадостійкій маніпуляції типу 4-ФМ (4-PSK), яка звичайно у технічній літературі трактується як квадратурна фазова маніпуляція – QPSK.

Назва маніпуляції цього й інших методів містить термін “квадратурна”. Можна відзначити, що квадратурне представлення сигналів є досить доцільним і потужним засобом описання цифрових модульованих сигналів.

Дійсно фазоманіпульоване коливання може бути зображено у вигляді лінійної комбінації синусоїдного і косинусоїдного коливань із нульовими початковими фазами. Це впливає із тригонометричного рівняння:

$$A(t) \cdot \cos(\omega_0 t + \varphi) = A(t) \cos \varphi \cdot \cos \omega_0 t - A(t) \sin \varphi \cdot \sin \omega_0 t =$$

$$= I(t) \cdot \cos\omega_0 t + Q(t) \sin\omega_0 t. \quad (2.4)$$

Відмітимо, що  $A(t)\cos\varphi$  і  $A(t)\sin\varphi$  на тактовому інтервалі є константами, і, значить, виступають коефіцієнтами під час представлення радіочастотного сигналу (2.4) у вигляді лінійної комбінації сигналів  $\cos\omega_0 t$  і  $\sin\omega_0 t$ . Оскільки складові  $\cos\omega_0 t$  і  $\sin\omega_0 t$  мають фазовий зсув  $90^\circ$ , то на фазовій діаграмі вони ортогональні (інакше кажучи, знаходяться у квадратурі).

Таким чином, для квадратурних каналів можливо розглядати дві окремих двійкових маніпуляції з їх наступною суперпозицією.

У такому представленні процеси модуляції і демодуляції набувають легшої схемотехнічної реалізації [4, 5]. На рис.2.3 зображено функціональну схему модулятора сигналів QPSK.

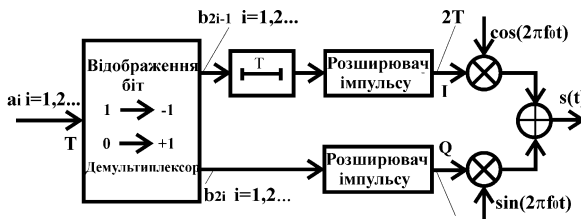


Рисунок 2.3 – Функціональна схема модулятора КФМ-4 (QPSK)

Такі модулятори отримали назву I, Q – модуляторів і для сигналу QPSK, який формується на виході модулятора, фазове сузір'я має вигляд квадратної решітки (рис. 2.4) з чотирма дискретними значеннями фази  $\varphi_i = \pm\pi/4, \pm3/4$  (стани A, B, C, D), а комплексна обвідна модульованого сигналу має постійне значення.

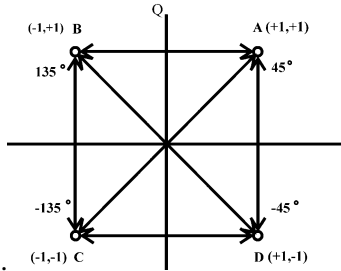


Рисунок 2.4 – Фазове сузір'я (діаграма фазових переходів) для сигналу КФМ-4 (QPSK)

Аналіз функціональної схеми рис. 2.3 свідчить, що у кожному каналі здійснюється бінарна фазова маніпуляція, але введення другого ортогонального каналу дозволяє практично удвічі збільшити ефективність використання заданої смуги частот, оскільки результуючий каналний символ вже переносить два інформаційних біти

## 2.2.2 Квадратурна амплітудна маніпуляція

Як видно з проведеного у параграфі 2.2.1 аналізу, при застосуванні М-ічної фазової модуляції амплітуда і частота несучого коливання впродовж сеансу зв'язку залишаються постійними. Змінюється тільки значення початкової фази кожного каналного символу.

Квадратурна амплітудна модуляція (quadrature amplitude modulation – QAM) передбачає одночасну зміну амплітуди і початкової фази кожного каналного символу. Один каналний символ для методу QAM може бути представленим у вигляді:

$$s(t) = A_m \cos(2\pi f_0 t + \Phi_m) = \operatorname{Re}[A_m \cdot \exp(j\Phi_m) \cdot \exp(j2\pi f_0 t)],$$

$$(i-1)T < t < iT,$$

де  $\dot{A}_m = A_m \cdot \exp(j\Phi_m)$  – комплексна амплітуда каналного символу,  $m=1, 2, \dots, M$ . Для побудови сигнального сузір'я доцільно представити вираз для  $s(t)$  у наступному вигляді:

$$s(t) = A_m \cos(2\pi f_0 t + \Phi_m) = A_m \cos(\Phi_m) \cos(2\pi f_0 t) - A_m \sin(\Phi_m) \sin(2\pi f_0 t) =$$

$$= a_m \cos(2\pi f_0 t) + b_m \sin(2\pi f_0 t), \quad (i-1)T < t < iT, \quad (2.5)$$

де коефіцієнти  $a_m$ ,  $b_m$  визначають координати  $m$ -ї точки фазового сузір'я.

Якщо порівняти вирази (2.4) для методу QPSK і вираз (2.5) для методів M-QAM, то можна зробити висновок, що якісно вони подібні і методи M-QAM можна також реалізувати на підставі квадратурного подання радіосигналу з використанням I, Q – модулятора.

Як було визначено один канальний символ M-QAM сигналу може переносити  $m = \log_2 M$  інформаційних бітів. Зокрема, для  $M = 16$  маємо  $m = 4$ . Тому, якщо вважати, що тривалість одного біта рівняється  $T_b$ , то тривалість одного канального символу QAM сигналу рівняється  $T_s = mT_b$ . Відповідно під час формування такого сигналу потік інформаційних бітів повинен групуватися у блоки з  $m$  бітів. Для кожного такого блоку повинен бути сформований один відповідний канальний радіосимвол. Процес встановлення такої відповідності в технічній літературі називають сигнальним кодуванням.

При великих значеннях  $M$  задавати множину можливих координат сигнальних точок найпростіше за допомогою цілих чисел, нумерація яких починають від початку координат.

Наприклад, для квадратної решітки, яку зображено на рис.2.4, можна ввести позначення  $a_{\min}$ ,  $b_{\min}$  для координат точок найближчих до початку координат. Тоді, якщо усі сусідні точки мають однакові відстані між собою впродовж кожної осі, то координати інших точок можна визначити через значення координат найближчих точок за допомогою співвідношень

$$a_k = \pm k a_{\min} \quad (2.6)$$

$$b_l = \pm l b_{\min}, \quad (2.7)$$

де індекси  $k$  і  $l$  приймають цілочислові значення.

Можна відзначити, що, в принципі, такий підхід і було застосовано під час формування символів для методу QPSK. Дійсно, метод QPSK можна трактувати не тільки як варіант методу 4-PSK, але й як окремий випадок методу QAM, а саме 4-QAM.

Якщо припустити, що для методу 4-QAM необхідно реалізувати сигнальне (фазове) сузір'я у вигляді квадратної решітки, то, очевидно, необхідно визначити координати двох точок по осі I і координати двох точок по осі Q.

При обранні мінімальної відстані по обом осям  $a_{\min}=1$ ,  $b_{\min}=1$ , то згідно з (2.6) і (2.7) ці дві координати для кожної осі відповідатимуть значенням коефіцієнтів  $k=1$  і складуть пару  $-1,+1$  по осі I і таку ж пару  $-1,+1$  по осі Q.

Оскільки один каналний символ M-QAM сигналу може переносити  $m=\log_2 M$  інформаційних бітів, то для  $M=4$  маємо  $m=2$ . Тому, якщо вважати, що тривалість одного біта рівняється  $T_b$ , то тривалість одного каналного символу 4-QAM сигналу рівняється  $T_s=2T_b$ . Відповідно під час формування 4-QAM сигналу потік інформаційних бітів повинен групуватися у блоки з 2 бітів. Для кожного такого блоку повинен бути сформований один відповідний каналний символ (рис. 2.5). Але саме так і формуються каналні символи для методу QPSK.

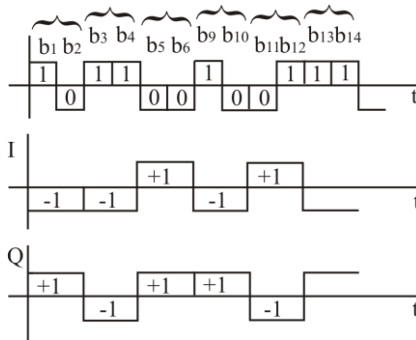


Рисунок 2.5 – Формування каналних символів для методу 4-QAM (QPSK)

Далі, якщо перетворення інформаційних біт на електричні імпульси відбувається за алгоритмом, для якого логічний “0” відповідає прямокутному імпульсу з рівнем  $+1$ , а логічна “1” відповідає прямокутному імпульсу з рівнем  $-1$ , то процедура сигнального кодування набуває простої інтерпретації – таблиця 2.1, а відповідне сигнальне сузір'я має вигляд квадратної решітки – рис.2.6.

Таблиця 2.1 – Сигнальне кодування методу 4-QAM (QPSK)

b	b	I	Q
0	0	1	1
0	1	1	-1
1	0	-1	1
1	1	-1	-1

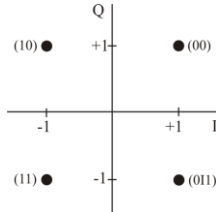


Рисунок 2.6 – Фазове сузір'я і сигнальне кодування для методу 4-QAM (QPSK)

Оскільки вирази і алгоритми для формування сигналів QPSK і M-QAM на підставі квадратурного подання сигналів подібні, то такий підхід до формування сигнального сузір'я можна розширити і на більші значення  $M > 4$ .

Дійсно, в [8] показано, що коефіцієнти  $k$  і  $l$ , які необхідні для визначення координат точок по осям I, Q, можна у загальному вигляді обчислити як:

$$k(l) = 2m - 1 - L, \quad (2.8)$$

де  $L = \sqrt{M}$ ,  $m = 1, 2, \dots, L$ .

Очевидно, що для розглянутого прикладу  $M = 4$ ,  $L = 2$  і для значень  $m = 1$ ,  $m = 2$  отримаємо множину точок  $-1, +1$ .

Якщо за запропонованим алгоритмом реалізувати сигнальне сузір'я у вигляді квадратної решітки для методу 16-QAM, то, очевидно, що для такого виду модуляції необхідно сформувати чотири значення для I компоненти і чотири значення для Q компоненти, що у підсумку дає 16 можливих станів.

Тобто, при  $M = 16$  і, відповідно,  $L = 4$ , з (2.8) для  $m = 1, m = 2, m = 3, m = 4$  отримуємо, що значення індексів  $k(l)$  належать множині  $(-3, -1, 1, 3)$ . Сукупність усіх точок цього сигнального сузір'я може бути задана за допомогою наступної матриці:

$$|k, l| = \begin{pmatrix} (-3,+3) & (-1,+3) & (+1,+3) & (+3,+3) \\ (-3,+1) & (-1,+1) & (+1,+1) & (+3,+1) \\ (-3,-1) & (-1,-1) & (+1,-1) & (+3,-1) \\ (-3,-3) & (-1,-3) & (+1,-3) & (+3,-3) \end{pmatrix}$$

Відповідно сигнальне кодування для методу 16-QAM відображено у таблиці 2.2, а фазове сузір'я наведено на рис.2.7.

Таблиця 2.2 – Сигнальне кодування для методу 16-QAM

B(1)	B(2)	B(3)	B(4)	I	Q	B(1)	B(2)	B(3)	B(4)	I	Q
0	1	0	1	3	3	1	0	0	1	-1	3
0	1	0	0	3	1	1	0	0	0	-1	1
0	1	1	0	3	-1	1	0	1	0	-1	-1
0	1	1	1	3	-3	1	0	1	1	-1	-3
0	0	0	1	1	3	1	1	0	1	-3	3
0	0	0	0	1	1	1	1	0	0	-3	1
0	0	1	0	1	-1	1	1	1	0	-3	-1
0	0	1	1	1	-3	1	1	1	1	-3	-3

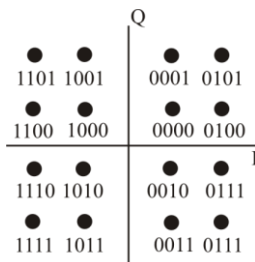


Рисунок 2.7 – Фазове сузір'я методу 16-QAM/

Сигнал може переходити до будь-якого стану наведеної фазової діаграми. Згідно з таблицею 2.2, для методу 16-QAM за один канальний символ передається чотири інформаційних біти. Це означає, що тривалість одного канального символу для цього методу  $T_c = 4T_b$ .

Відповідно потік інформаційних біт для методу 16-QAM згруповується у блоки з чотирьох біт і кожному такому блоку відповідає один канальний символ.

Необхідно відзначити, що різні канальні символи сигналу 16-QAM (на відміну від методу QPSK) мають різну енергію, відстань між різними сигнальними точками також відрізняється. В результаті ймовірність невірної визначення символу у приймачі для різних символів також виходить різною.

Квадратурна амплітудна модуляція при  $M=16$  знайшла застосування у транкінговій системі радіозв'язку iDEN (інтегрована цифрова мережа з розширеними можливостями), високошвидкісних WPAN мережах за стандартом IEEE 802.15.3, цифрових релейних станціях і т.ін..

Запропонований алгоритм формування фазового сузір'я у вигляді квадратної решітки застосовується і при більш складних методах QAM (64-QAM, 256-QAM).

Наприклад, для  $M = 64$  і  $L = 8$  відповідно до (2.8) отримаємо, що значення індексів  $k(l)$  належать множині  $(-7, -5, -3, -1, 1, 3, 5, 7)$ , що надає вісім координат по осі I і вісім координат по осі Q, що в результаті надає 64 точки на фазовому сузір'ї. Відповідне фазове сузір'я для 64-QAM сумісно з результатами сигнального кодування зображено на рис.2.8.

Наведені на рис.2.6...2.8 фазові сузір'я мають форму квадратної решітки, у вузлах якої розташовані сигнальні точки. Можна відзначити, що це не єдина можлива форма фазового сузір'я і не завжди оптимальна. Зокрема, сигнальне сузір'я може бути реалізованим у вигляді хреста, кола і т. ін., що особливо доцільно для більших значень  $M$  [11...13]. У сучасних системах зв'язку значення  $M$  може перевищувати 1024.

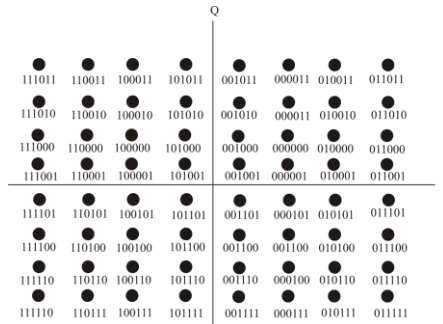


Рисунок 2.8 – Фазове сузір'я для методу 64-QAM

Так, наприклад, на рис.2.9 зображено фазове сузір'я для варіанту 16-QAM з квадратною решіткою (рис.2.9а) і варіанту QAM зі зниженою чутливістю до флуктуацій фази несучої частоти (рис.2.9б).

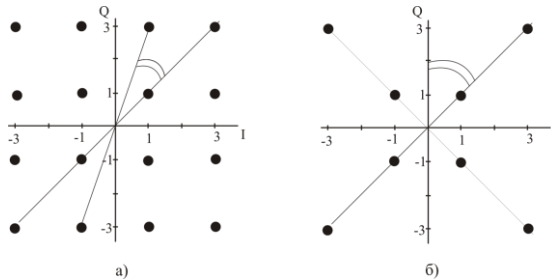


Рисунок 2.9 – Варіанти сигнального сузір'я для методів M-QAM

Зниження чутливості досягається завдяки більшому куту між сигнальними векторами. Характерною особливістю сузір'я на рис. 2.9 б є те, що модулюючі параметри для I, Q компоненти взаємозв'язані і не можуть обиратися незалежно один від одного. Наприклад, значення  $I = 1$  може відповідати лише двом значенням другого модулюючого параметру  $Q = \pm 1$ .

Якщо порівняти спектральну ефективність методу M-QAM з іншими видами цифровими методами модуляції, зокрема, з бінарними, то можна відзначити, що M-QAM має найкращі показники з цього параметру. Дійсно, було показано, що для M-рівневої багатопозиційної модуляції ( $M = 2^k$ ) відбувається формування каналних символів тривалістю  $T_s = kT_b$ . Це означає, що швидкість передавання інформації при використанні M-рівневих модуляцій збільшується у k-разів. В таблиці 2.3 наведені порівняльні характеристики бінарних та M-QAM модуляцій, які свідчать про значно більшу спектральну ефективність методів M-QAM.

Таблиця 2.3 – Порівняльні характеристики простих і багатопозиційних видів модуляції

Показник	BPSK	QPSK	16-QAM	64-QAM
Теоретична спектральна ефективність	1	2	4	6
Реальна спектральна ефективність в системах з лінійним підсиленням, біт/(с·Гц)	0,8	1,6	3,2	5
Реальна спектральна ефективність в системах з нелінійним підсиленням, біт/(с·Гц)	0,3	0,7	1...1,5	2...2,5
S/N, дБ, що необхідне для $P_b=10^{-4}$	8,4	11,4	19,2	25,5
$E_b/N_0$ , дБ, що необхідне для $P_b=10^{-4}$	8.4	8,4	13,2	17,8

З аналізу таблиці 2.3 можна зробити висновок, що для досягнення підвищеної спектральної ефективності або зниження необхідного значення S/N при заданій ймовірності похибки, або і того і другого разом, доцільно застосовувати методи M-QAM. Якщо канал (приймач/передавач) вважається лінійним, то за допомогою

багатопозиційної модуляції можна наблизитися до теоретичної границі Шенона.

Ширина спектра такого сигналу приблизно така ж як і для  $M$ -ічного фазоманіпульованого сигналу. Але даний спосіб модуляції може забезпечити меншу ймовірність бітової помилки і тому у ряді випадків застосування  $M$ -QAM є більш доцільним (рис.2.10 [6]).

У загальному випадку  $M$ -рівневі системи  $M$ -QAM з лінійним підсиленням мають спектральну ефективність більшу порівняно з методом QPSK, який має теоретичну граничну ефективність 2 біт/(с·Гц), а, наприклад, система 256-QAM з лінійним підсиленням має теоретичну спектральну ефективність 8 біт/(с·Гц).

Для порівняння можна відзначити, що метод модуляції GMSK, який набагато ефективніший з точки зору перетворення енергії джерела живлення у потужність радіочастотного сигналу, має значно меншу спектральну ефективність (приблизно 1 біт/(с·Гц)).

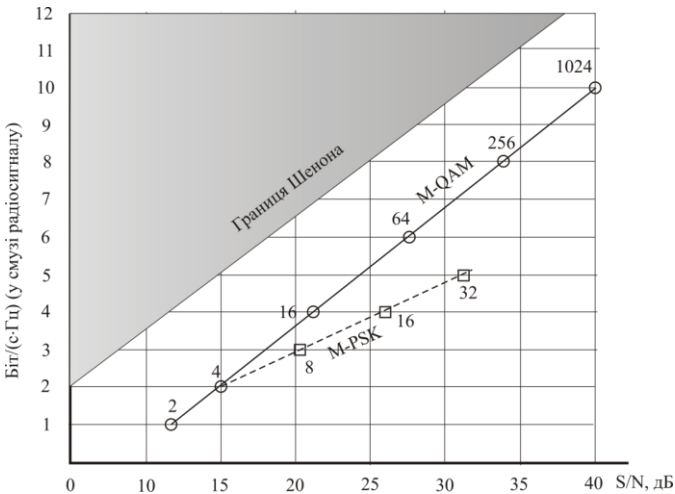


Рисунок 2.10 – Залежність спектральної ефективності  $M$ -позиційної когерентної  $M$ -QAM і  $M$ -PSK від відношення  $S/N$  при  $P_b=10^{-8}$

Але, з іншого боку,  $M$ -QAM сигнал не має постійної амплітуди, а це означає, що застосування цього способу модуляції вимагає підвищених вимог до лінійності каналу передачі. Дійсно, практичне

застосування методів M-QAM зустрічається з дуже важливою проблемою міжсимвольної інтерференції, що за умови впливу нелінійностей як передавального, так і приймального тракту, призводить до заміни одного символу на інший, тобто збільшення ймовірності бітової помилки.

Тому, такі види модуляції, як 16-QAM, 64-QAM, 256-QAM вимагають застосування ультралінійних підсилювачів і забезпечення значно більшого відношення сигнал/шум порівняно з більш простими методами QPSK і GMSK (рис. 2.11) [6].

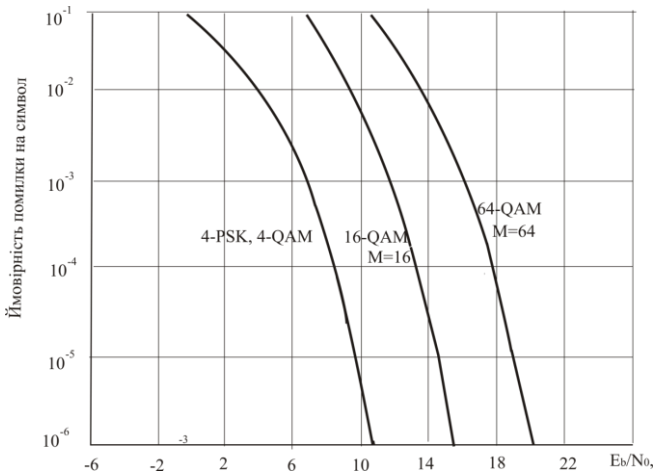


Рисунок 2.11 – Залежність ймовірності помилки на символ від відношення  $E_b/N_0$

Так, для досягнення ймовірності бітової помилки  $P_b = 10^{-4}$  метод 16-QAM порівняно з QPSK вимагає приблизно на 8 дБ більшого значення відношення сигнал/шум (19,2 дБ замість 11,4 дБ) і збільшення на 4,8 дБ відношення енергії, що приходить на один біт, до спектральної щільності шуму –  $E_b/N_0$  (13,2 дБ замість 8,4 дБ)[6].

Тому відзначимо, що, незважаючи на більш високі показники з теоретичної спектральної ефективності, використання QAM через підвищення вимог до відношення сигнал/шум не дозволяє реально

збільшити спектральну ефективність, наприклад, системи стільникового зв'язку у цілому.

З цієї причини квадратурна амплітудна модуляція на теперішній час у системах стільникового зв'язку не використовується. Але, методи M-QAM знайшли широке застосування у стандартах персональних, локальних і регіональних безпроводних мереж.

На підставі проведеного аналізу можна запропонувати схему модулятора M-QAM сигналів на базі виразу (2.5) і розглянутої схеми I, Q модулятора (рис. 2.12).

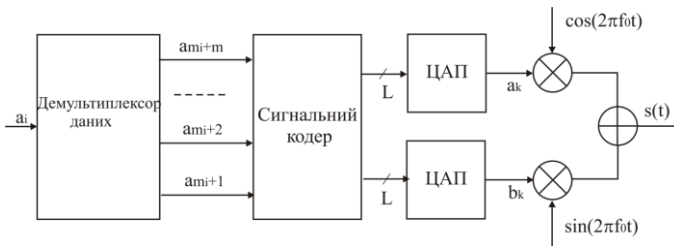


Рисунок 2.12 – Функціональна схема модулятора M-QAM сигналу

Відповідно, спрощена схема каналу зв'язку з використанням M-QAM зображена на рис. 2.13 [9].

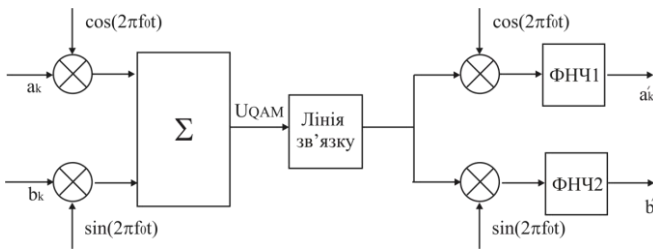


Рисунок 2.13 – Структурна схема каналу зв'язку з використанням M-QAM

При моделюванні необхідно взяти до уваги той факт, що згідно з теоремою Найквіста про відсутність міжсимвольних спотворень необхідно застосувати для каналних символів формуючий фільтр Найквіста.

На практиці фільтр, який задовольняє вимозі теореми Найквіста про часткову симетрію, отримав назву фільтр типу “піднесений косинус”. Імпульсна характеристика такого фільтра визначається виразом [9]:

$$h(t) = \frac{1}{T} \cdot \frac{\sin(t/T)}{t/T} \frac{\cos[2\pi(B-f_0)t]}{[1-4(B-f_0)t]^2} \quad (2.9)$$

де  $T$ - тривалість символу,  $f_0=1/2T$ ,  $B$ -смуга пропускання фільтра.

Амплітудно-частотну характеристику фільтра типу “піднесений косинус” можна подати у вигляді :

$$A(f) = 1, \quad |f| < f_0(1-r);$$

$$A(f) = \sqrt{\frac{1}{2} + \frac{1}{2} \sin\left[\frac{\pi}{2f_0} \left(\frac{f_0 - |f|}{\alpha}\right)\right]}, \quad f_0(1-\alpha) \leq |f| \leq f_0(1+\alpha);$$

$$A(f) = 0, \quad |f| > f_0(1+\alpha); \quad (2.10)$$

Відповідні імпульсна характеристика та амплітудно-частотна характеристика фільтра зображені на рис.2.14.

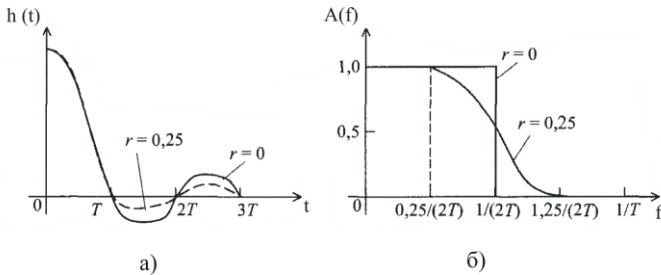


Рисунок 2.16 – Імпульсна (а) та амплітудно-частотна характеристики фільтра типу “піднесений косинус”

Параметром характеристики, який визначає розширення смуги пропускання фільтра відносно частоти Найквіста,  $\epsilon$ , так званий, коефіцієнт згладжування  $r$ . Цей коефіцієнт визначається як:

$$r = (B - f_0) / f_0.$$

Наприклад, фільтр, що використовується в стандарті WiMax має коефіцієнт згладжування  $r=0,25$ .

### 3 АЛГОРИТМ РОБОТИ СИГНАЛЬНОГО КОДЕРА

На підставі функціональної схеми модулятора і демодулятора QAM необхідно сформувані структурні схеми передавального і приймального трактів моделі каналу зв'язку з використанням M-QAM і визначити основні задачі, які виносяться на моделювання.

Як впливає з проведеного аналізу, радіосигнал з M-QAM модуляцією формується шляхом додавання радіосигналів, які формуються у двох незалежних квадратурних I, Q каналах. Формування модулюючих каналних I, Q символів реалізується на базі сигнального кодера (рис. 2.12).

Алгоритм роботи сигнального кодера передбачає виконання таких процедур:

а) проводиться розділення вхідної послідовності біт ( $a_{2m+1}, \dots, a_{2m+2n}$ ) на дві підпослідовності:  $b_1(a_{2m+1}, \dots, a_{2m+n})$  і  $b_2(a_{2m+n+1}, \dots, a_{2m+2n})$ .

б) Для обох підпослідовностей здійснюються наступні обчислення:

$$a_I(k) = \sum_{j=1}^n a_{2m+j} \times 2^{n-j} \quad (3.1)$$

$$a_Q(k) = \sum_{j=1}^n a_{2m+n+j} \times 2^{n-j} \quad (3.2)$$

в) проводиться обчислення значень I і Q каналних символів. Для цього використовується матриця перетворення C, де коефіцієнти матриці розраховані відповідно до (2.8) [1]:

$$C = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3 & -1 & 1 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -7 & -5 & -1 & -3 & 7 & 5 & 1 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -15 & -13 & -9 & -11 & -1 & -3 & -5 & -7 & 15 & 13 & 9 & 11 & 1 & 3 & 5 & 7 \end{pmatrix}, \quad (3.3)$$

Амплітуди значень I і Q каналних символів обчислюються за такими виразами:

$$d_I(k) = C[n, a_I(k)+1] / \sum_{j=1}^n 2^{j-1} \quad (3.4)$$

$$d_Q(k) = C[n, a_Q(k)+1] / \sum_{j=1}^n 2^{j-1}, \quad (3.5)$$

де  $n$  – порядковий номер рядка матриці  $C$  відповідно до методу M-QAM, що моделюється,  $[a_I(k)+1], [a_Q(k)+1]$  визначають порядковий номер елемента в обраному рядку матриці.

Параметр  $n$  відповідає поданню методу M-QAM у вигляді  $2^{2n}$  – QAM. Це означає, що наведена матриця може застосовуватися для різних видів M-QAM, оскільки значення  $n=1,2,3,4$  визначають методи 4-QAM, 16-QAM, 64-QAM, 256-QAM відповідно.

Можна відзначити, що запропонований алгоритм є загальним для реалізації будь-яких методів  $2^{2n}$  - QAM модуляції. Наприклад, для  $n=1$ , тобто для метода 4-QAM, перша процедура призводить до розділення вхідної послідовності символів на підпослідовність непарних ( $a_1, a_3, \dots, a_{2m+1}$ ) та парних ( $a_2, a_4, \dots, a_{2m+2}$ ) символів.

Відповідно для 4-QAM обчислення за виразами (3.1), (3.2) для перших чотирьох бітів надає наступне:

– з виразу (3.1), за умови  $n=1, m=0$  перший I-каналний символ  $a_I(1)$  визначається безпосередньо значенням  $a_1$ , а другий I-каналний символ ( $n=1, m=1$ )  $a_I(2)$  визначається значенням  $a_3$ , тобто  $a_I(1) = a_1, a_I(2) = a_3$ ;

– з виразу (3.2) виходить, що перший Q-каналний символ  $a_Q(1)$  визначається значенням  $a_2$ , а другий символ  $a_Q(2)$  визначається значенням  $a_4$ , тобто  $a_Q(1) = a_2, a_Q(2) = a_4$ .

Тобто, цим алгоритмом практично реалізується розбиття вхідної послідовності на блоки з двох інформаційних бітів, причому перший

біт (непарний) з блоку надходить до I-каналу, а другий (парний) – до Q- каналу.

Далі, відповідно до (3.4), (3.5) з матриці перетворення  $C$  використовується перший рядок і згідно зі значенням  $a_{1(Q)}+1$  з нього обирається безпосередньо значення модулюючого символу. Можна відзначити, що, оскільки коефіцієнти матриці мають як від'ємний, так і додатний знаки, то при такому алгоритмі відпадає необхідність додаткової процедури перетворення інформаційної послідовності до формату NRZ. Процес реалізації сигнального кодування для 4-QAM за таким алгоритмом роботи продемонстровано на рис.3.1.

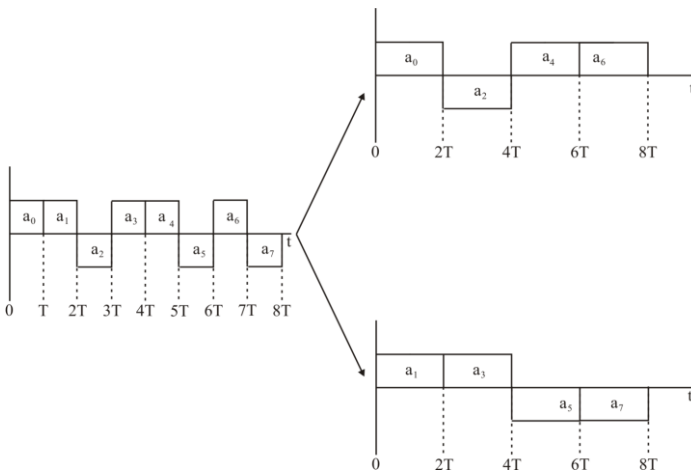


Рисунок 3.1 – Алгоритм формування каналних символів для методу 4- QAM

Для формування сигнального сузір'я застосуємо вирази (3.4), (3.5) і матрицю перетворення (3.3).

Очевидно, що для  $n=1$ , використовується тільки перший рядок матриці  $C$  і тоді каналні символи отримуються згідно з (3.4), (3.5) відповідно. Так, для значення  $a_1(1) = 0$  з (3.4) виходить, що перший I-каналний символ  $d_1(1) = C(1,1) = -1$  і, відповідно, перший Q-каналний символ також визначиться як  $d_Q(1)=C(1,1)= -1$ . Очевидно, якщо  $a_1(k)= 1$  і  $a_Q(k)=1$ , то значення каналних символів будуть визначатися другим елементом першого рядка матриці  $C$ , тобто  $d_1(k)=C(1,2)= +1$  і  $d_Q(k)=C(1,2)= +1$ . В результаті ці процедури

призведуть до формування наступних координат I, Q символів для методу 4-QAM (таблиця 3.1).

Таблиця 3.1 – Сигнальне кодування для методу 4-QAM

Непарний	Парний біт	I	Q
0	0	-1	-1
0	1	-1	1
1	0	1	-1
1	1	1	1

Якщо змінити порядок знаків у матриці перетворення на протилежний, тобто, використовувати матрицю C у вигляді:

$$C = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 1 & -1 & -3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 7 & 5 & 1 & 3 & -7 & -5 & -1 & -3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 15 & 13 & 9 & 11 & 1 & 3 & 5 & 7 & -15 & -13 & -9 & -11 & -1 & -3 & -5 & -7 \end{pmatrix},$$

то результат сигнального кодування визначиться таблицею 3.2.

Таблиця 3.2 – Сигнальне кодування для методу 4-QAM (другий варіант)

Непарний біт	Парний біт	I	Q
0	0	1	1
0	1	1	-1
1	0	-1	1
1	1	-1	-1

Обидва можливі варіанти фазового сузір'я зображені на рис.3.2.

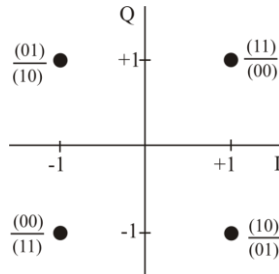


Рисунок 3.2 – Сигнальне сузір'я методу 4-QAM

Очевидно, що отримане сигнальне сузір'я повністю тотожне сигнальному сузір'ю класичного методу QPSK. Як відомо, послідовність інформаційних бітів для методу QPSK спочатку перетворюється до формату NRZ без постійної складової.

Причому, якщо для перетворення застосовувати алгоритм, для якого логічна одиниця “1” перетворюється на електричний імпульс з рівнем +1, а логічний нуль “0” відповідає електричному імпульсу з рівнем -1, то отримаємо перший варіант сигнального сузір'я (на рис.3.2 значення дибітів відповідають чисельнику). Якщо ж логічній одиниці відповідає електричний імпульс з рівнем -1, а логічному нулю електричний імпульс з рівнем +1, то формується другий варіант сигнального сузір'я (на рис.3.2 значення дибітів відповідають знаменнику).

Розглянемо процес формування каналних символів для методу 16-QAM, Подаючи 16-QAM у вигляді  $2^{2n}$  – QAM, очевидно, що у даному випадку  $n=2$ . Тоді за першою процедурою відбувається розбиття вхідної послідовності на блоки з чотирьох бітів, причому перші два біти надходять до I каналу, а третій і четвертий до Q каналу. Відповідно з виразів (3.1), (3.2) для коефіцієнтів  $a_I(k)$ ,  $a_Q(k)$  отримаємо:

$$a_I(1) = 2a_1 + a_2, \quad (3.6)$$



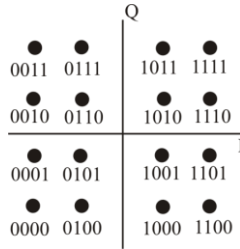


Рисунок 3.3 – Сигнальне сузір'я методу 16-QAM

Аналізуючи рис.3.3 і, зокрема, значення блоків з чотирьох бітів для сусідніх точок сигнального сузір'я, необхідно відзначити, що запропонований алгоритм кодування реалізує, так зване, кодування Грея. За таким кодуванням сусідні точки сигнального сузір'я відрізняються значенням тільки одного біта.

При переході до моделювання методу  $2^{2 \cdot 3}$  - QAM, тобто 64 – QAM, за першою процедурою відбувається формування блоків з шести інформаційних бітів, причому перші три біти надходять до I каналу, а четвертий, п'ятий і шостий до Q каналу. Відповідно з виразів (3.1), (3.2) для коефіцієнтів  $a_I(k)$ ,  $a_Q(k)$  отримаємо:

$$a_I(1) = 4a_1 + 2a_2 + a_3, \quad (3.8)$$

$$a_Q(1) = 4a_4 + 2a_5 + a_6. \quad (3.9)$$

Формування безпосередньо амплітуд I, Q – канальних символів проводиться за (3.4), (3.5) з використанням третьої строки матриці перетворення. Очевидно, що далі цей процес можливо розповсюдити і на методи 256 – QAM, 1024- QAM з використанням матриці перетворення більш високого порядку.

## 4 МОДЕЛЬ КАНАЛУ ЗВ'ЯЗКУ З ВИКОРИСТАННЯМ МОДУЛЯЦІЇ М-QAM

### 4.1 Структурна схема моделі каналу зв'язку з використанням методу QAM

Після формування каналних символів I, Q символів у сигнальному кодері для кожного з каналів модулятора подалі здійснюється APSK (амплітудно-фазова) маніпуляція.

Сучасні модеми каналів зв'язку реалізуються, як правило, на базі так званої DSPH – архітектури, тобто з використанням цифрових сигнальних процесорів (DSP) і, відповідно, цифрових методів обробки сигналів.

Тому, після формування каналних символів у сигнальному кодері, в модуляторі здійснюється операція передискретизації, а саме на часовому інтервалі, що відповідає тривалості одного символу проводиться додавання нульових відліків згідно з обраним порядком дискретизації.

Для забезпечення передачі інформації в каналі зв'язку без міжсимвольної інтерференції необхідно для модулюючої послідовності застосувати фільтрацію з використанням формуючого фільтра Найквіста.

Відповідно в кожному з каналів модулятора QAM після формування каналних I, Q символів у сигнальному кодері і здійснення операції передискретизації проводиться подальша обробка модулюючого сигналу у формуючому фільтрі з АЧХ та імпульсною характеристикою, що визначаються виразами (2.9), (2.10).

Для реалізації безпосередньо амплітудно-фазової модуляції в кожному з каналів необхідно перетворити сигнал до аналогової форми за допомогою цифро-аналогового перетворювача (ЦАП).

Результуючі сигнали після перемножувачів і є вихідними сигналами кожного з каналів, які подаються до суматора, на виході якого утворюється модульований M-QAM радіосигнал для передавання через канал зв'язку.

Відповідна структурна схема передавального тракту з використанням методу M-QAM за запропонованим алгоритмом зображена на рис. 4.1.

Структурна схема приймального тракту для методу M-QAM зображена на рис. 4.2. На приймальному боці прийнятий антеною сигнал, по-перше, проходить через смуговий фільтр, який забезпечує придушення усіх позасмугових завадових сигналів.

Подальша обробка сигналу здійснюється за допомогою I,Q демодулятора, який як і I,Q модулятор складається з двох квадратурних каналів.

Першим кроком обробки у кожному з каналів виступає перенесення радіосигналу до основної смуги частот. Для моделювання пропонується схема когерентного прийому, для якої на приймальному боці формується поновлене несуче коливання. Відповідно для I-каналу таким коливанням виступає  $K\cos\omega_0t$ , а для квадратурного каналу  $K\sin\omega_0t$ .

В результаті перемноження прийнятого сигналу і поновлених квадратур після ФНЧ утворюється сигнал в основній смузі частот, з якого необхідно виділити інформаційну послідовність.

У приймальному тракту усі подальші кроки виконуються також за допомогою DSPH апаратури із застосуванням цифрових методів обробки сигналів. Відповідно сигнал знову необхідно подати у вигляді цифрових вибірок, для чого застосовується аналогово-цифровий перетворювач (АЦП). Аналогічно, як і в модуляторі, у кожному каналі демодулятора застосовується формуючий цифровий фільтр, як і на передавальному боці.

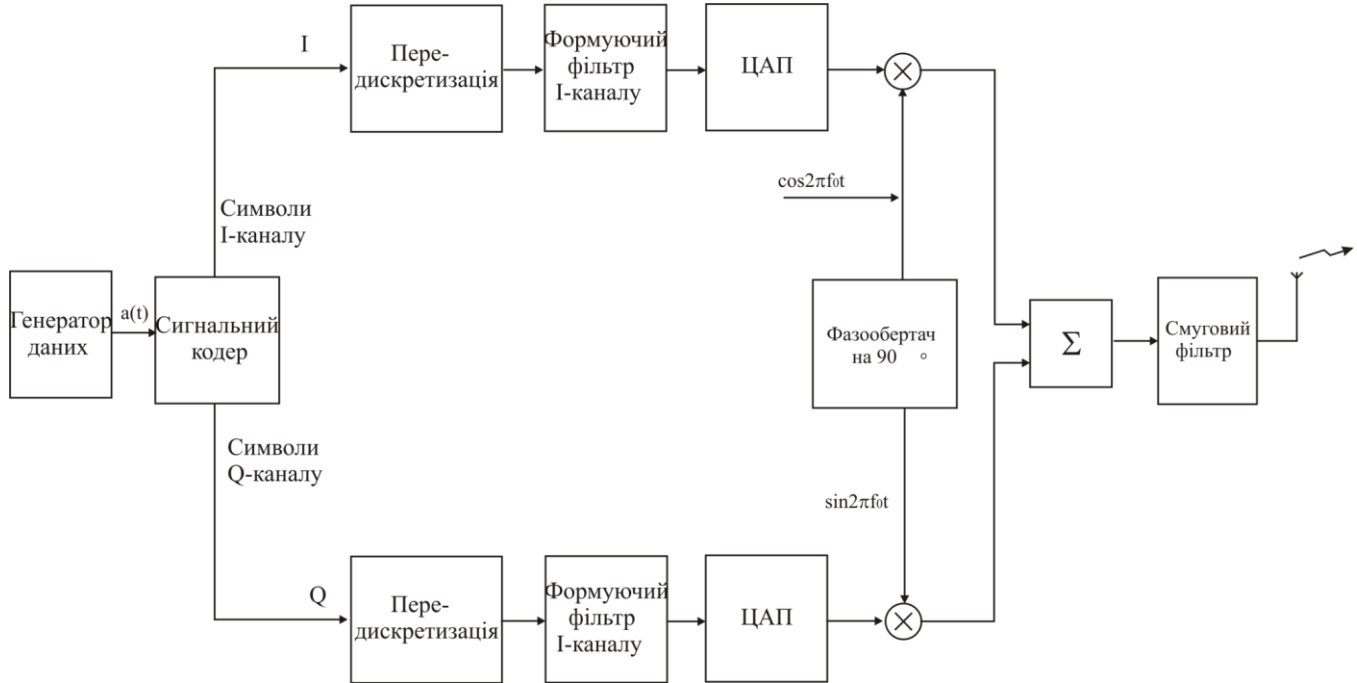


Рисунок 4.1 – Структурна схема передавального тракту каналу зв'язку з використанням методу QAM

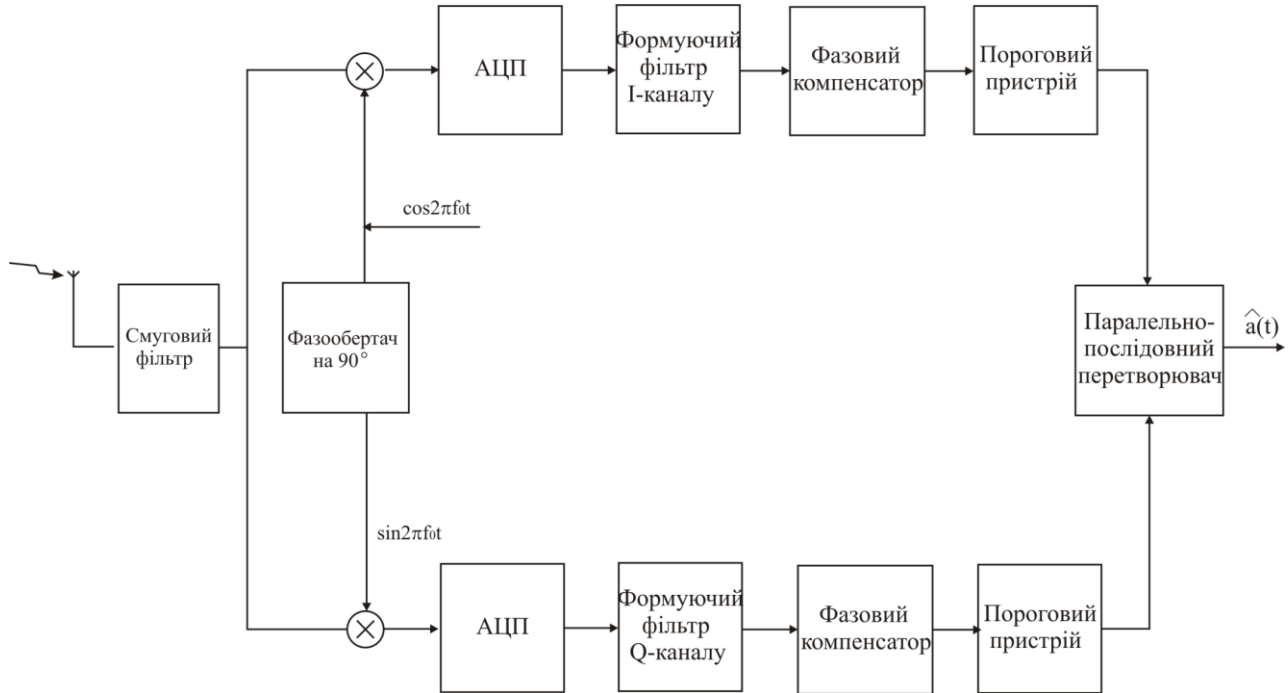


Рисунок 4.2 – Структурна схема приймального тракту каналу зв'язку з використанням методу QAM

Для відфільтрованого дискретизованого сигналу необхідно обрати точку синхронізації і провести операцію зворотної пере дискретизації.

#### 4.2 Реалізація програмного алгоритму для AWGN каналу з QAM модуляцією

У відповідності до проведеного у параграфі 4.1 аналізу, на рис.4.3 зображена блок-схема алгоритму моделювання модему.

Сама програмна модель складається з головної програми **main.m**, яка передбачає порядок виконання усіх етапів обробки сигналу у відповідності до структурних схем рис.4.1, рис.4.2 і блок-схеми рис.4.3 і шести підпрограм, які власне, реалізують відповідні види обробки сигналу.

Для забезпечення моделювання процесу цифрової модуляції за методом M-QAM необхідний такий набір підпрограм: `comproversamp.m`, `compconv.m`, `hrollcoef.m`, `qammod.m`, `comb.m`, `qamdmod.m`.

Підпрограми `comproversamp.m`, `compconv.m` реалізують операцію передискретизації і згортки передискретизованої послідовності з імпульсною характеристикою формуючого фільтра, коефіцієнти якої наведені у підпрограмі `hrollcoef.m`.

За допомогою підпрограми `comb.m` до вхідного сигналу демодулятора додається білий шум визначеної потужності. Принцип дії цих програм та їх функціональні можливості достатньо детально проаналізовані в [3, 4].

Підпрограми `qammod.m` `qamdmod.m` реалізують процедури сигнального кодування і сигнального декодування відповідно до заданого методу M-QAM.

Повні лістинги головної програми і необхідних підпрограм наведені у додатку А методичних вказівок.

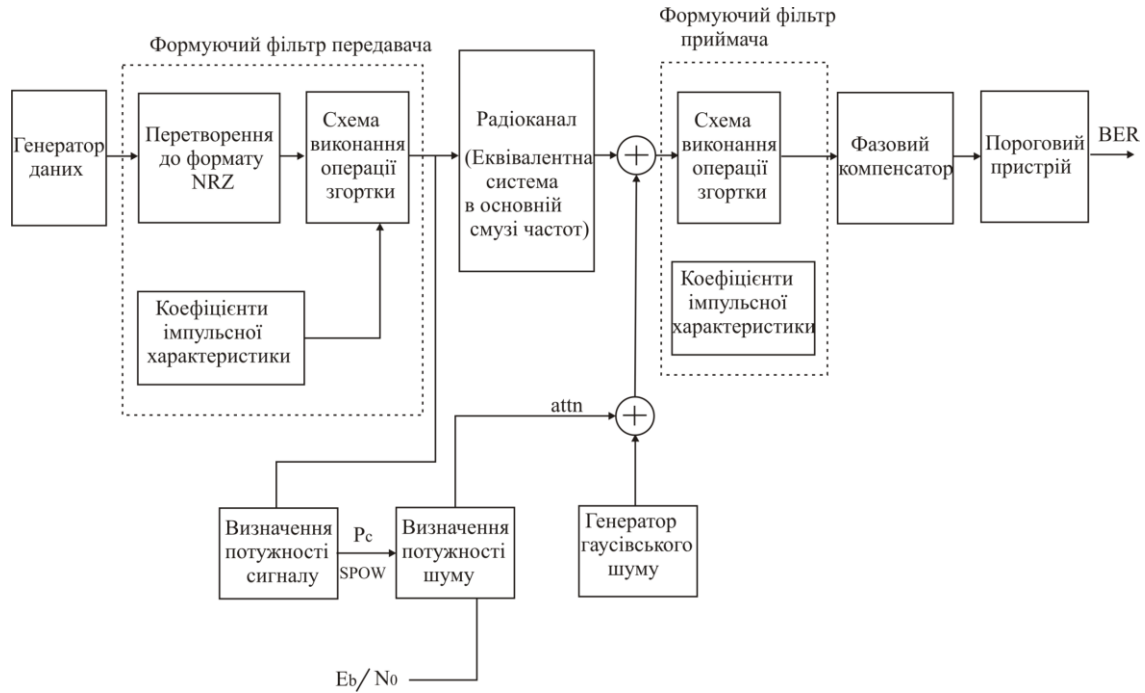


Рисунок 4.3 – Блок-схема алгоритму моделювання модема QAM

### 4.3 Опис основної програми

По-перше вводяться початкові дані для моделювання. Введемо такі позначення:

sr [кбіт/с] – швидкість передачі символів;

br [кбіт/с] – швидкість передачі інформаційних бітів;

ml – кількість бітів, що передаються за один символ. Як відомо, кількість бітів, що переносяться за один символ визначається виразом:  $ml = \log_2 M$ , де  $M$  – основа обраної системи числення. Наприклад, якщо аналізується 16-QAM ( $M=16$ ), то очевидно, що для такого варіанту необхідно задати  $ml=4$ ;

nd – кількість переданих символів, для яких буде проводиться розрахунок ймовірності помилкового прийому за одну реалізацію випадкового процесу (один цикл моделювання) ;

ebn0 – початкове значення енергії, що приходить на один біт  $E_b$ , до спектральної щільності шуму  $N_0$ . Як відомо, у цифрових системах зв'язку аналогм відношення сигнал/шум, яке характеризує аналогові системи, виступає саме параметр  $E_b/N_0$ , оскільки для цифрової системи необхідно враховувати швидкість передавання інформаційних бітів у визначеній смузі частот.

IPOINT – порядок передискретизації.

Наприклад, для методу 16-QAM при швидкості передачі інформації 256 кбіт/с блок введення початкових даних у термінах алгоритмічної мови програмування Matlab набуває наступного вигляду:

```
% main.m
% Simulation program to realize QAM transmission system
%*****Preparation part*****
sr=256000.0; % Symbol rate
ml=4; % ml : Number of modulation levels
br=sr.*ml; % bit rate
nd=100; % Number of symbols
ebn0=3; % Eb/N0
IPOINT=8; % Number of oversamples.
```

Кількість бітів ml, що переносяться за один радіосимвол, як було визначено, для  $2^{2n}$  – QAM рівняється  $2n$  і задається у програмі параметром  $ml = 4$ , відповідно бітова швидкість br визначається як результат перемноження символної швидкості на параметр ml, що у

випадку QAM-16 означає, що бітова швидкість рівняється  $ml*nd$ . Тобто у випадку 16-QAM бітова швидкість для наведеного прикладу рівняється 1024 кбіт/с.

Кількість символів, що підлягають аналізу для однієї реалізації випадкового процесу прийнята рівною  $nd = 100$ , початкове значення параметру  $E_b/N = 3$ .

Наступним блоком програми є завдання параметрів формуючого фільтра Найквіста. Оскільки формуючий фільтр застосовується як на передавальному так і на приймальному боці, то на практиці застосовується фільтр відомий як “корінь квадратний з фільтра Найквіста” або фільтр типу  $\sqrt{RC}$ . Для моделювання відклику такого фільтра на вхідну послідовність даних необхідно визначити імпульсну перехідну характеристику такого фільтра.

Коефіцієнти імпульсної характеристики фільтра  $\sqrt{RC}$  розраховуються за допомогою підпрограми **hrollfcoef(irfn, IPOINT,sr,alfs,1/0)**.

Параметрами підпрограми є кількість ліній затримки фільтра **irfn**, раніше визначені порядок передискретизації **IPOINT** і символічна швидкість **sr**, а також коефіцієнт скруглення **alfs**. Для моделювання кількість ліній затримки прийнята рівною **irfn = 21**, а значення коефіцієнта округлення **alfs = 0.5**

Подалі починається основне тіло програми.

Введемо наступні позначення:

**nloop** – кількість циклів моделювання;

**nod** – кількість переданих бітів;

**poe** – кількість помилково декодованих бітів на приймальній стороні.

Для того, щоб адекватно визначити імовірносні характеристики каналу зв'язку задамо кількість циклів моделювання (кількість реалізацій випадкового процесу) достатньо великою – приймаємо значення **nloop=1000**. Початкові значення кількості переданих і помилково декодованих бітів приймаються рівними нулю.

Змінну цикла позначимо через **iii** а початок циклу задається оператором **for iii=1:nloop**. Відповідний фрагмент програми набуває наступного вигляду:

```
%*****Start calculations*****
nloop=1000; % Number of simulation loops
```

```

noe=0;          % Number of error data
nod=0;          % Number of transmitted data
for iii=1:nloop
....
Тіло циклу
...
end

```

По-перше, генерується випадкова інформаційна послідовність усі елементи якої обираються з ансамблю  $\{0,1\}$ . Вхідні дані задаються у вигляді вектора розмірністю  $1 \cdot nd \cdot ml$  за допомогою наступної команди:

```

%*****Data generation*****
data=rand(1,nd*ml)>0.5 % rand: built in function

```

Наступним кроком відповідно до блок-схеми (рис. 2.6) є формування послідовності модулюючих символів I-каналу – вектор [ich] та Q-каналу – вектор [qch] згідно з алгоритмом роботи сигнального кодера, роботу якого розглянуто у параграфі 2.1. Цей алгоритм реалізується за допомогою підпрограми qammod:

```

% qammod.m
% Function to perform QAM modulation
function [iout,qout]=qammod(paradata, para, nd, ml).
Змінні підпрограми qammod такі:
% paradata – послідовність вхідних даних
% iout – вихідна послідовність каналних символів I – каналу;
% qout – вихідна послідовність каналних символів Q – каналу;;
% para – кількість паралельних каналів;
% nd – кількість вхідних даних;;
% ml – кількість бітів , що переносяться за один радіо символ

```

Повий текст підпрограми наведено нижче :

```

%***** QAM mapping*****
iv=[ -1 1 ];
m2=ml./2;
paradata2=paradata.*2-1;
count2=0;
for jj=1:nd
    isi=zeros(para,1);
    isq=zeros(para,1);
    for ii=1:m2
        isi=isi + 2.^(m2-ii).*paradata2((1:para), ii+count2);
        isq=isq + 2.^(m2-ii).*paradata2((1:para), m2+ii+count2);
    end
end

```

```

end
    iout((1:para),jj)=isi;
    qout((1:para),jj)=isq;
    count2=count2+ml;
end

```

Оскільки цю підпрограму планується використовувати для будь-якого методу M-QAM, де тривалість символу  $T_c = \log_2 M \cdot T_b$ , то в підпрограмі організовано додатковий цикл (змінна циклу- **ii**), який передбачає формування блоків з 2, 4, 6 бітів для методів 4-QAM, 16-QAM, 64-QAM і т.д.

Наприклад, для методу 4-QAM цей цикл виконується тільки один раз для кожного каналу і формує підпоследовності парних **isi** і непарних **isq** символів з вхідної последовності.

Вибір необхідного рядка з матриці C реалізується першим оператором підпрограми:

```
iv=[ -1 1 ];
```

У наведеному прикладі підпрограмою **qammod.m** моделюється метод 4-QAM. Відповідно з матриці C обрано перший рядок і його подано вектором **iv** (для узгодження розмірів матриць нульові коефіцієнти не задаються). Для реалізації іншого методу M-QAM необхідно задати у векторі **iv** відповідний рядок матриці C (тільки не нульові елементи).

Для методу 16-QAM згаданий цикл виконується два рази для кожного каналу і в результаті формуються символи I-каналу **isi** і каналні символи Q-каналу **isq** за допомогою виразів (3.1),...(3.5) (з урахуванням того, що вектор **iv** вже містить чотири коефіцієнти з другого рядка матриці C).

Так, наприклад, якщо для вхідної последовності відліків першим формується блок з чотирьох бітів :  $[a_1=1, a_2=0, a_3=0, a_4=1]$ , то перші два біти надходять до I- каналу, а третій і четвертий до Q-каналу.

В результаті роботи підпрограми за (3.1), (3.2) для першого блоку отримаємо:

$$a_I(1)=2a_1+a_2=2 \cdot 1+0=2$$

$$a_Q(1)=2a_3+a_4=2 \cdot 0+1=1$$

Далі з матриці  $C$  обирається другий рядок і коефіцієнти цього рядка задаються у вигляді вектору  $\mathbf{iv}=[-3 \ -1 \ 1 \ 3]$  у підпрограмі **qammod.m**, а значення каналних символів визначаються за (3.3), (3.4), (3.5):

$$C_2=[-3 \ -1 \ 1 \ 3],$$

$$d_i(1)=C[2, a_i(1)+1]=C[2,3]= +1,$$

$$d_q(1)=C[2, a_q(1)+1]=C[2,2]= -1.$$

Результатом роботи програми **qammod.m** є формування двовірного вектора **[ich,qch]**, який відповідає комплексній послідовності  $I, Q$  – каналних символів.

Для сформованого вектора **[ich,qch]** проводиться операція передискретизації, а саме на часовому інтервалі, що відповідає тривалості одного символу, проводиться додавання нульових відліків згідно з обраним порядком дискретизації. Порядок дискретизації згідно з технічним завданням задано рівним  $IPOINT=8$ .

Операція передискретизації виконується за допомогою підпрограми `compoversamp(data1,nd,IPOINT)` (додаток А):

**[ich1,qch1]=compoversamp(ich,qch,length(ich),IPOINT);**

Змінним підпрограми виступають вектори **ich** і **qch**, довжина цих векторів **length(ich)** і порядок передискретизації **IPOINT**.

На рис. 4.4 показано операцію передискретизації більш детально (у збільшеному масштабі) для перших можливих п'яти символів  $I$ -каналу для методу 16-QAM.

Сформовані відліки у відповідності з функціональною схемою модулятора підлягають обробці у формуючому фільтрі Найквіста. Як відомо сигнал на виході будь-якого фільтра можна отримати за допомогою операції згортки вхідного сигналу і імпульсної характеристики фільтра. Для реалізації згортки застосовується вбудована функція Mathlab **conv** (convolution):

В головній програмі операція фільтрації в кожному каналі реалізується за допомогою за допомогою підпрограми **compconv.m**. Результатом роботи програми виступає двовірний вектор **[ich2,qch2]**, який формує комплексну послідовність відфільтрованих каналних символів **[ich2,qch2]** з комплексної

послідовності  $[\mathbf{ich1}, \mathbf{qch1}]$ . Змінними підпрограми виступають вектори  $\mathbf{ich1}$  і  $\mathbf{qch1}$  і коефіцієнти імпульсної характеристики формуючого фільтра  $\mathbf{xh}$ .

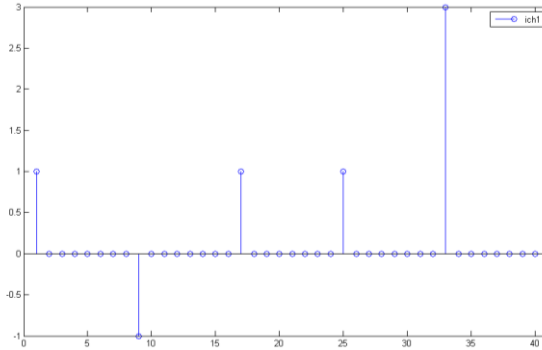


Рисунок 4.4 – Демонстрація операції передискретизації

Відповідно три підпрограми, які, практично формують вихідний сигнал QAM модулятора в основній смузі частот набувають такого вигляду:

```
%*****QAM Modulation*****
[ich,qch]=qammod(data1,1,nd,ml);
[ich1,qch1]=compoversamp(ich,qch,length(ich),IPOINT);
[ich2,qch2]=compconv(ich1,qch1,xh);
```

Для здійснення безпосередньо APSK модуляції в кожному каналі необхідно цифровий сигнал перетворити до аналогової форми за допомогою ЦАП.

Оскільки моделювання процесу перенесення на робочу частоту не передбачалося технічним завданням, тобто, модель розглядається в основній смузі частот, то на цьому етапі модель модулятора можна вважати сформованою.

Процес обробки сигналів на приймальному боці в моделі демодулятора починається з визначення відношення сигнал/шум для прийнятого сигналу у припущенні, що розглядається AWGN канал.

Введення шуму до переданого сигналу відповідно до заданого значення  $E_b/N_0$  реалізується за допомогою алгоритму, що розглянуто в [3, 4] на підставі виразу

$$P_\phi = \frac{P_c}{F} \cdot \frac{R}{10^{\frac{E_b/N_0}{10}}}, \quad (4.13)$$

де  $P_c$  – потужність сигналу;

$R$  – швидкість передачі символів;

$F$  – частота слідкування інформаційних бітів.

Потужність передавання одного символу  $P_c$  (параметр SPOW), визначається як середньоквадратична сума потужностей усіх символів I і Q – каналів, яка поділена на кількість символів.

Враховуючи, що для загального варіанту застосування I, Q модуляторів, шум однаково розподілений в I та Q каналах модулятора, остаточно вводиться шумова напруга  $attn$  як

$$attn = \sqrt{\frac{1}{2} \cdot P_\phi}.$$

В програмі цей алгоритм реалізовано наступними операторами

```
%*****Attenuation Calculation*****
spow=sum(ich2.*ich2+qch2.*qch2)/nd;
attn=0.5*spow*sr/br*10.^(-ebn0)/10);
attn=sqrt(attn); % sqrt: built in function.
```

Додавання гаусівського шуму до корисного сигналу відповідно до заданого значення  $E_b/N_0$  здійснюється за допомогою підпрограми `comb.m`. В підпрограмі `comb.m` процедура введення шуму реалізується для обох каналів демодулятора з наступним додаванням обчислених шумів до корисного сигналу в кожному каналі демодулятора:

В результаті вектори даних, які формують передані через канал символи (в основній смузі частот) на вході I- каналу [ich3] і Q- каналу [qch3] демодулятора реалізуються оператором:

```
[ich3, qch3]= comb (ich2, qch2, attn); % add write gaussian noise.
```

Подальша обробка символів в кожному з каналів демодулятора передбачає фільтрацію у формуючому фільтрі з формуванням вектора **[ich4,qch4]** за допомогою підпрограми **compconv.m**, [3, 4]:

```
[ich4,qch4]=compconv(ich3,qch3,xh2);
```

Вибір точки синхронізації і усунення передискретизації відліків в I і Q каналах повністю тотожні алгоритму, якого розглянуто в [3, 4]:

```
syncpoint=irfn*IPOINT+1;
```

```
ich5=ich4(syncpoint:IPOINT:length(ich4));
```

```
qch5=qch4(syncpoint:IPOINT:length(qch4));
```

В результаті на виходах I і Q- каналів демодулятора формуються двополярні послідовності демодульованих каналних символів **ich5** і **qch5** відповідно (можливий вигляд демодульованих символів зображено на рис.4.5, 4.6).

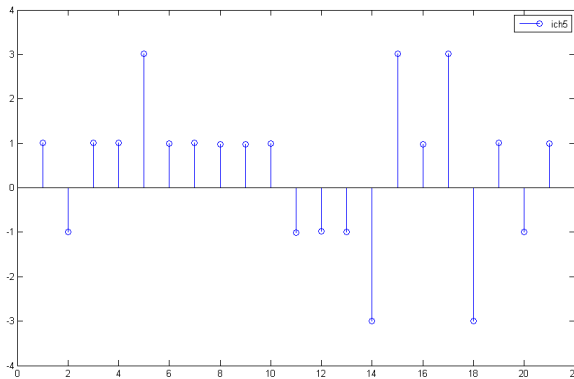


Рисунок 4.5 – Демодульована послідовність символів I-каналу для методу 16-QAM

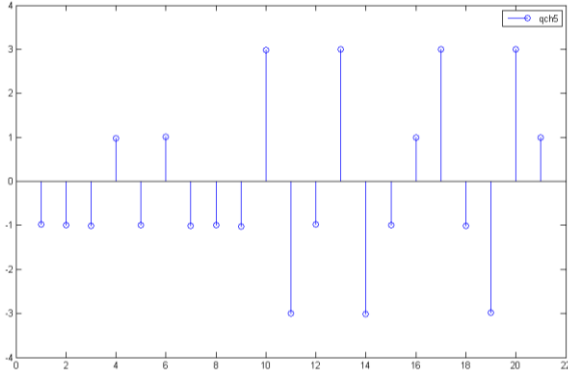


Рисунок 4.6 – Демодульована послідовність символів Q-каналу для методу 16-QAM

Результат демодуляції сигналів у вигляді однополярної послідовності відліків формується за допомогою підпрограми **qamdemod.m**:

```
%*****QAM Demodulation*****
[demodata]=qpskdemod(ich5,qch5,1,nd,ml);
```

Підпрограма **qamdemod.m** практично реалізує алгоритм роботи порогових пристроїв в I, Q каналах демодулятора і реалізує функцію сигнального декодера.

Розроблена програмна модель дозволяє також відобразити і фазові сузір'я для різних значень  $E_b/N_0$ . Для цього достатньо задати вивід графіка, де по осі абсцис буде відкладається значення демодульованих двополярних відліків I-каналу, а по осі ординат значення демодульованих відліків Q-каналу. Отримана діаграма фазових переходів за наявності гаусівських шумів у каналі зв'язку буде суттєво відрізнятися від теоретичної, особливо для малих значень  $E_b/N_0$ .

Програмний комплекс дозволяє провести моделювання для методів 4-QAM, 16-QAM, 64-QAM, 256-QAM.

Різниця у головній програмі під час моделювання за допомогою розробленого комплексу полягає тільки у введенні відповідного

значення змінної  $ml$ – яка визначає довжину сформованого блоку у бітах (або кількість бітів, що передаються за один радіосимвол):

<b>ml=2</b>	для методу	4-QAM;
<b>ml=4</b>	для методу	16-QAM;
<b>ml=6</b>	для методу	64-QAM;
<b>ml=8</b>	для методу	256-QAM.

#### 4.4 Алгоритм розрахунку бітової помилки

Для розрахунку ймовірності помилкового прийому одного біта необхідно визначити відношення кількості помилково прийнятих бітів до загальної кількості переданих бітів. Для розрахунку кількості помилок запропоновано наступний алгоритм.

Спочатку проводиться віднімання від однойменних бітів переданої вхідної послідовності даних (data) відповідних бітів демодульованої послідовності на приймальному боці (demodata).

Якщо помилок на приймальному боці не виявлено, то отримаємо нульовий вектор довжиною “nod”. У випадку ж з’явлення помилково декодованих бітів формується вектор довжиною “nod”, в якому на позиціях, що відповідають помилково декодованим бітам з’являться значення +1 або -1.

Застосувавши операцію виділення абсолютного значення можна отримати вектор, який матиме одиничні значення на помилкових позиціях. Якщо тепер провести операцію арифметичного додавання усіх елементів сформованого вектора, то отримаємо загальну кількість помилок, які виникли під час демодуляції. Визначимо розраховану за таким алгоритмом кількість помилок як змінну “noe”

Після проведення визначених процедур можна розрахувати ймовірність помилкового прийому одного біта як відношення кількості помилково прийнятих бітів до загальної кількості переданих бітів. В термінах Matlab, послідовність операцій така:

```
%*****Bit Error Rate (BER)*****
noe2=sum(abs(data-demodata));
    % sum: built in function
nod2=length(data1);    % length: built in function
noe=noe+noe2;
```

```

nod=nod+nod2;
end % for iii=1:nloop
%*****Output result *****
ber=noe/nod;

```

Для розглянутої у параграфі головної програми **main.m** вивід результатів, щодо ймовірності помилкового прийому буде виводитися тільки для одного конкретного значення  $E_b/N_0$ .

Для формування графічних залежностей BER необхідно використовувати іншу головну програму, яка є модернізацією програми **main.m**.

З метою формування графічної залежності  $BER=F(E_b/N_0)$  перед основним циклом (оператор **for iii=1:nloop**) створюється додатковий цикл за параметром  $E_b/N_0$ .

З цією метою вводяться три нових масиви:

1) **ebn0\_array** – масив значень  $E_b/N_0$  від одного до 40 дБ з кроком 1 дБ;

2) **ber\_array** – масив значень BER, який визначається за результатами моделювання для відповідних значень  $E_b/N_0$ ;

3) **ber\_array\_theory** – масив значень BER, який визначається з теоретичного виразів для відповідних методів M-QAM;

Відповідно цикл реалізовано такими операторами:

```

ebn0_array=0:1:25;
ber_array=zeros(length(ebn0_array),1);
ber_array_theory=zeros(length(ebn0_array),1);
for ebn0_loop=1: length(ebn0_array) ;
ebn0= ebn0_array (ebn0_loop) ;

```

Для виводу результатів щодо BER необхідно після основного циклу визначити розраховані масиви помилок, а саме:

```

ber_array(ebn0_loop)=ber;
ber_array_theory(ebn0_loop)=0.5*erfc(sqrt(10^(ebn0_array
(ebn0_loop)/10)));
end % for ebn0_loop

```

Графік необхідно вивести у напівлогарифмічному масштабі, тобто для коефіцієнта BER застосувати логарифмічний масштаб, а для

$E_b/N_0$  застосувати лінійний масштаб у дБ. Для цього застосуємо такі оператори:

```
h=semilogy(ebn0_array, ber_array,'o', ebn0_array, ber_array_theory);  
set(h,{'DisplayName'},{'Experiment','Theory'});  
legend show
```

За базову залежність під час графічної репрезентації поведінки BER для різних методів M-QAM приймемо криву, яка характеризує класичний метод QPSK. Теоретичний вираз для ймовірності помилкового прийому одного біта або коефіцієнта бітової помилки методу QPSK для AWGN каналу має вигляд [6, 9]:

$$BER_{QPSK,AWGN} = \frac{1}{2} \operatorname{erfc}(\sqrt{E_b / N_0}). \quad (4.1)$$

Як було показано, метод 4-QAM практично тотожний методу QPSK і тому порівнювати залежності BER для інших методів M-QAM будемо саме з методом QPSK. Саме виразу (4.1) відповідає оператор

```
ber_array_theory(ebn0_loop)=0.5*erfc(sqrt(10^(ebn0_array...  
(ebn0_loop)/10)));;
```

Теоретичний вираз для коефіцієнта бітової помилки методу 16-QAM для AWGN каналу має вигляд [9]:

$$BER_{16QAMAWGN} = \frac{3}{8} \operatorname{erfc}(\sqrt{\frac{2}{5} E_b / N_0}) - \frac{9}{64} \operatorname{erfc}^2(\sqrt{\frac{2}{5} E_b / N_0}). \quad (4.2)$$

Відповідні теоретичні вирази для ймовірності помилкового прийому одного біта для AWGN каналу у варіанті застосування методів 64-QAM і 256-QAM наведені нижче [9]:

$$BER_{64QAMAWGN} = \frac{7}{24} \operatorname{erfc}(\sqrt{\frac{1}{7} E_b / N_0}) - \frac{49}{384} \operatorname{erfc}^2(\sqrt{\frac{1}{7} E_b / N_0}). \quad (4.3)$$

$$\text{BER}_{256\text{QAM AWGN}} = \frac{15}{64} \operatorname{erfc}\left(\sqrt{\frac{4}{85}} E_b/N_0\right) - \frac{225}{2048} \operatorname{erfc}^2\left(\sqrt{\frac{4}{85}} E_b/N_0\right). \quad (4.4)$$

Наведені вирази (4.1)...(4.4) свідчать, що значення коефіцієнта бітової помилки BER для різних методів M-QAM при одних і тих же значеннях відношення  $E_b/N_0$  будуть суттєво відрізнятися.

Цей факт стає очевидним, якщо провести порівняння фазового сузір'я для методу 16-QAM і, отриманого за одних і тих же умов в каналі, фазового сузір'я для методу 256-QAM (рис. 4.7).

Очевидно, що відстань між сусідніми точками сузір'я для 256-QAM виходить суттєво меншою, що під час впливу шумів значно ускладнює задачу прийняття рішення сигнальним декодером приймача щодо відповідності прийнятого сигналу конкретній точці фазового сузір'я.

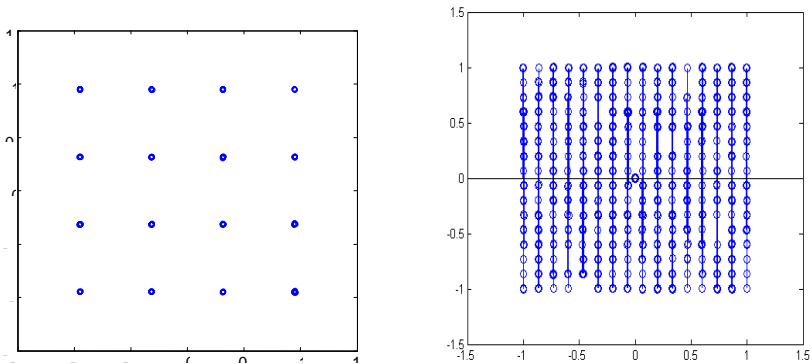


Рисунок 4.14 – Нормовані фазові сузір'я методів 16-QAM і 256-QAM

## 5 ПОРЯДОК ВИКОНАННЯ КУРСОВОЇ РОБОТИ

### 5.1 Обґрунтування алгоритму роботи сигнального кодера

Варіанти технічного завдання приведені у таблиці 5.1. Відповідно до заданого у технічному завданні методу M-QAM провести процедури визначення модулюючих канальних I,Q символів:

5.1.1 Визначити кількість бітів  $K$ , для якої відбувається розбиття вхідної послідовності на блоки з  $K$  бітів.

Для цього представити задані у технічному завданні методи модуляції у вигляді:

$$M\text{-QAM} \leftrightarrow 2^{2n} - \text{QAM} \quad (5.1)$$

Відповідно показник степені  $K=2n$  визначає кількість бітів для одного блоку.

5.1.2 Визначити тривалість одного канального символу  $T_s$  і кількість бітів, які переносяться одним радіосимволом для відповідного методу M-QAM:

$$T_s = T_b \log_2 K .$$

Кількість бітів, які будуть переноситися одним радіосимволом рівняється  $K$ .

5.1.3 Провести в аналітичному вигляді виділення з вхідної послідовності перших двох блоків з  $K$  бітів для відповідного методу M-QAM. Для цього вхідну послідовність подати у вигляді  $(a_{2m+1}, \dots, a_{2m+2n})$ , де значення  $m=0, n, 2n, 3n, \dots$  послідовно визначають номер відповідного блоку. Записати сформовані два блоки для відповідного методу у вигляді:

Метод  $2^{2n}$  - QAM

Перший блок  $(a_{2m+1}, \dots, a_{2m+2n}) \big|_{m=0} = a_1, \dots$

Другий блок  $(a_{2m+1}, \dots, a_{2m+2n}) \big|_{m=n} = a_{2n+1}, \dots$

Таблиця 5.1 – Завдання на курсову роботу

Варіант	Послідовність для проведення сигнального кодування	Метод модуляції	Швидкість передавання (символ/с)	Коефіцієнт округлення	Порядок передискретизації
1	0001001010100001	16-QAM	128	0,3	8
2	0000110000101110	64-QAM	256	0,5	8
3	1011001010100111	256-QAM	128	0,25	8
4	1011001010110011	16-QAM	256	0,4	8
5	1011001011100111	64-QAM	128	0,3	8
6	1111111101111111	256-QAM	256	0,5	8
7	0110001010100111	16-QAM	256	0,35	8
8	1110001010000111	64-QAM	300	0,3	8
9	1111000011110100	256-QAM	360	0,5	8
10	1100011010100111	4-QAM	256	0,25	8
11	1011101010100111	16-QAM	128	0,25	8
12	1100001110000111	64-QAM	256	0,5	8
13	1010011110110011	256-QAM	128	0,3	8
14	0100010110100111	16-QAM	256	0,4	8
15	0111000111100111	64-QAM	128	0,3	8
16	1000000010000001	256-QAM	312	0,5	8
17	0011001010100111	4-QAM	360	0,25	8
18	1111110000000100	64-QAM	240	0,3	8
19	1111111100000000	256-QAM	512	0,5	8
20	0000111100001111	16-QAM	256	0,25	8

5.1.4 Визначити кількість бітів, які формують один субблок у кожному каналі I, Q модулятора.

Подання M-QAM у вигляді (5.1) безпосередньо визначає кількість біт n, що утворюють один окремий субблок в I та Q каналах модулятора з вхідної послідовності даних.

5.1.5 Провести розділення вхідної послідовності біт ( $a_{2m+1}, \dots, a_{2m+2n}$ ) на дві підпослідовності, які формуються з послідовних субблоків:  $b_I(a_{2m+1}, \dots, a_{2m+n})$  і  $b_Q(a_{2m+n+1}, \dots, a_{2m+2n})$ .

Записати сформовані два субблоки для відповідного методу у вигляді:

Метод  $2^{2n}$  - QAM

Перший субблок I каналу  $b_{1I}(a_{2m+1}, \dots, a_{2m+n}) \big|_{m=0} =$

Другий субблок I каналу  $b_{2I}(a_{2m+1}, \dots, a_{2m+n}) \big|_{m=n} =$

Перший субблок Q каналу  $b_{1Q}(a_{2m+n+1}, \dots, a_{2m+2n}) \big|_{m=0} =$

Другий субблок Q каналу  $b_{2Q}(a_{2m+n+1}, \dots, a_{2m+2n}) \big|_{m=n} =$  .

5.1.6 Для обох підпослідовностей виконати наступні обчислення:

$$a_I(k) = \sum_{j=1}^n a_{2m+j} \times 2^{n-j}, \quad (5.2)$$

$$a_Q(k) = \sum_{j=1}^n a_{2m+n+j} \times 2^{n-j} \quad (5.3)$$

де k – визначає номер субблоку, з якого формується аналітичний вираз для коефіцієнтів  $a_I(k)$  і  $a_Q(k)$ .

5.1.7 Провести обчислення значень I і Q каналних символів. Для цього необхідно застосувати матрицю перетворення C:

$$C = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3 & -1 & 1 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -7 & -5 & -1 & -3 & 7 & 5 & 1 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -15 & -13 & -9 & -11 & -1 & -3 & -5 & -7 & 15 & 13 & 9 & 11 & 1 & 3 & 5 & 7 \end{pmatrix}$$

Для заданого методу оброти відповідний рядок матриці (значення  $n$  у поданні (5.1) визначає рядок матриці). Записати обрані рядки матриці відповідно до завдання.

5.1.8 Для заданої у завданні послідовності здійснити сигнальне кодування для перших двох блоків відповідно до заданого методу M-QAM.

Для реалізації кодування необхідно до виразів, які отримані у пп. 5.1.3...5.1.7 підставити значення інформаційних бітів і отримати чисельні значення коефіцієнтів  $a_I(k)$  і  $a_Q(k)$ . За визначеними значеннями обчислити амплітуди  $I$  і  $Q$  канальних символів за виразами:

$$\begin{aligned}d_I(k) &= C[n, a_I(k)+1], \\d_Q(k) &= C[n, a_Q(k)+1],\end{aligned}$$

де  $n$ - порядковий номер рядка матриці  $C$  відповідно до методу M-QAM, що моделюється, а значення  $[a_I(k)+1]$ ,  $[a_Q(k)+1]$  визначають порядковий номер елемента в обраному рядку матриці.

## 5.2 Експериментальна частина

5.2.1 Відкрити середовище Matlab. Створити нову директорію KURSOVA, в яку необхідно розмістити основну програму, а також усі підпрограми, що необхідні для моделювання.

5.2.2 Скопіювати файл основної програми **main.m**, який знаходиться у додатку А методичних вказівок. Зберегти скопійований файл під ім'ям **main.m** у створеній директорії KURSOVA.

5.2.3 Скопіювати у додатку А підпрограму **qammod.m**, і повторити процедуру створення нового М-файлу в директорії KURSOVA.

5.2.4 Скопіювати у додатку А підпрограму **hrollcoef.m**, і повторити процедуру створення нового М-файлу в директорії KURSOVA.

5.2.5 Скопіювати у додатку А підпрограму **compoversamp.m**, що виконує операцію передискретизації, і повторити процедуру створення нового М-файлу в директорії KURSOVA.

5.2.6 Скопіювати у додатку А підпрограму **compconv.m**, що виконує операцію згортки, і повторити процедуру створення нового М-файлу в директорії KURSOVA.

5.2.7 Скопіювати у додатку А підпрограму **comb.m**, що виконує операцію додавання шуму, і повторити процедуру створення нового М-файлу в директорії KURSOVA.

5.2.8 Скопіювати у додатку А підпрограму **qamdemod.m**, і повторити процедуру створення нового М-файлу в директорії KURSOVA.

Після виконання вище визначених пунктів, в основному вікні середовища Matlab усі 7 М-файлів повинні відобразитися у полі поточної директорії (Current Directory).

5.2.9 У командній строчці вікна Command Window набрати назву основної програми **main** і запустити аналіз натисканням клавіші Enter.

За результатами моделювання отримати наступні епюри:

- 1) вхідну послідовність даних у вигляді цифрових відліків;
- 2) підпослідовність цифрових символів І-каналу;
- 3) підпослідовність цифрових символів Q-каналу;
- 4) підпослідовність цифрових відліків І-каналу після передискретизації;
- 5) підпослідовність цифрових відліків Q-каналу після передискретизації;
- 6) відфільтровану підпослідовність цифрових відліків І-каналу;
- 7) відфільтровану підпослідовність цифрових відліків Q-каналу;
- 8) модулюючу послідовність І-каналу (після ЦАП) ;
- 9) модулюючу послідовність Q-каналу (після ЦАП) ;
- 10) сигнал основної смуги частот на вході І-каналу демодулятора ;
- 11) сигнал основної смуги частот на вході Q-каналу демодулятора ;
- 12) сигнал основної смуги частот на виході формуючого фільтра І-каналу демодулятора ;
- 13) сигнал основної смуги частот на виході формуючого фільтра Q- каналу демодулятора ;
- 14) демодульовані канальні символи І-каналу;
- 15) демодульовані канальні символи Q-каналу;
- 16) фазове сузір'я

5.2.10 Змінити значення параметру  $E_b/N_0$  в головній програмі на значення  $E_b/N_0=10$  дБ. Повторити розрахунок. Зафіксувати нове фазове сузір'я.

5.2.11. Повторити розрахунки при значеннях  $E_b/N_0=20$  дБ і  $E_b/N_0=30$  дБ. Зафіксувати отримані фазові сузір'я.

5.2.12 Отримати графічну залежність ймовірності помилкового прийому біта (або коефіцієнта BER) від відношення  $E_b/N_0$  для заданого у завданні методу M-QAM.

Для цього необхідно скопіювати у додатку Б нову головну програму **QAM\_BER.m** і зберігти її у директорії KURSOVA. Провести у ній необхідні зміни, щодо виводу результатів теоретичного аналізу, відповідно до заданого методу M-QAM.

Зміни полягатимуть у наступному. Після рядка, який визначає теоретичну залежність BER для методу QPSK (4-QAM):

```
ber_array_theory(ebn0_loop)=0.5*erfc(sqrt(10^(ebn0_array(ebn0_loop)/10))),
```

необхідно ввести нову змінну **ber\_array\_theory1(ebn0\_loop)** і визначити її відповідно до заданого методу згідно з (4.2)...(4.4).

```
ber_array_theory1(ebn0_loop)= ...
```

Після цього, у блоці виводу результатів аналізу замість операторів

```
h=semilogy(ebn0_array,ber_array,'o',ebn0_array,ber_array_theory,'--');  
set(h,{'DisplayName'},{'Experiment','Theory'})
```

необхідно задати оператори

```
h=semilogy(ebn0_array,ber_array,'o',ebn0_array,ber_array_theory,'--',  
...ebn0_array,ber_array_theory1,'*');  
set(h,{'DisplayName'},{'Experiment','Theory','Theory1'}),
```

що дозволить вивести результати залежності BER за результатами експерименту і теоретичними виразами для методу 4-QAM (QPSK) і заданого методу M-QAM. Зберігти файл натисканням кнопки Save.

5.2.13 У командній строчці вікна Command Window набрати назву нової основної програми **QAM\_BER** і запустити аналіз натисканням клавіші Enter.

Графічну залежність подати у напівлогарифмічному масштабі з діапазоном зміни ймовірності від  $10^{-5}$  до 1.

5.2.14 Порівняти залежності отримані за результатами моделювання і теоретичними виразами, а також провести порівняння характеристик при значеннях  $BER=10^{-3}$  і  $BER=10^{-4}$ .

**ПЕРЕЛІК ПОСИЛАНЬ**

1. Климаш М.М. Проектування ефективних систем безпроводного зв'язку [Текст] / М.М. Климаш, В.О. Пелішок. – Львів, 2010. – 232 с.
2. Борисов. В.И. Помехозащищенность систем радиосвязи с расширением спектра сигналов методом псевдослучайной перестройки частоты [Текст] / В.И. Борисов, В.М. Зинчук, А.Е. Лимарев; под.ред. В.И. Борисова. – М.: РадиоСофт, 2008. – 512 с.
3. Ломовицкий В.В. Основы построения систем и сетей передачи информации [Текст] / В.В. Ломовицкий, А.И. Михайлов, К.В. Шестаков, В.М. Щекотихин. – М.: Горячая линия, 2005. – 382 с.
4. Кабак В.С. Функціональні пристрої телефонів мобільного зв'язку: навчальний посібник [Текст] / В.С. Кабак, Р.В. Уваров. – Запоріжжя, 2007. – 375 с.
5. Маковеева М.М. Системы связи с подвижными объектами [Текст] / М.М. Маковеева., ШинаковЮ.С. – М.: Радио и связь, 2002. – 440 с.
6. Феер К. Беспроводная цифровая связь [Текст] / К. Феер; пер. с англ.; – М.: Радио и связь, 2000. – 519 с.
7. Скляр Б. Цифровая связь. Теоретические основы и практическое применение [Текст] / Б. Скляр; пер. с англ. – М.: Издательский дом “Вильямс”, 2003. – 1104 с.
8. Theodor S. Rappaport. Wireless Telecommunications. Principles and practice. – Prentice Hall
9. Harada H. Simulation and software radio for mobile telecommunications [Текст] / H. Harada, R. Prasad. – Artech House, 2003. – 465 p.

## ДОДАТОК А

Лістинги програм для моделювання M-QAM

```

% main
% Simulation program to realize 4QAM transmission system
% ***** preparation part *****
sr=256000.0; % Symbol rate
ml=2; % ml : Number of modulation levels
% (BPSK : ml=1, QPSK : ml=2, 16QAM : ml=4)
br=sr.* ml; % Bit rate
nd = 1000; % Number of symbols that simulates in each loop
ebn0=8.5; % Eb/N0
IPOINT=8; % Number of oversamples
% ***** Filter initialization *****
irfn=21; % Number of taps
alfs=0.5; % Rolloff factor
[xh] = hrollfcoef (irfn,IPOINT,sr,alfs,1); %Transmitter filter coefficients
[xh2] = hrollfcoef (irfn,IPOINT,sr,alfs,0); %Receiver filter coefficients
% ***** START CALCULATION *****
nloop=100; % Number of simulation loops
noe = 0; % Number of error data
nod = 0; % Number of transmitted data
for iii=1:nloop
% ***** Data generation *****
data1=rand (1, nd*ml) > 0.5 ;
% ***** 4 QAM Modulation *****
[ich qch] = qammod (data1, 1, nd, ml);
[ich1 qch1]=compoversamp (ich, qch, length (ich), IPOINT);
[ich2 qch2]=compconv (ich1, qch1, xh);
% ***** Attenuation Calculation *****
spow=sum(ich2.* ich2+ qch2.*qch2) / nd;
attn=0.5*spow*sr/br*10.^ (-ebn0/10);
attn=sqrt (attn);
% ***** Add Write Gaussian Noise (AWGN) *****
[ich3, qch3]= comb (ich2, qch2, attn) ; % add write gaussian noise
[ich4, qch4]= compconv (ich3, qch3, xh2) ;
saml=irfn*IPOINT+1;
ich5 = ich4 (saml : IPOINT : length (ich4)) ;
qch5 = qch4 (saml : IPOINT : length (ich4)) ;
% ***** 4QAM Demodulation *****

```

```

[demodata]=qamdemod (ich5, qch5,1,nd,ml);
% ***** Bit Error Rate (BER) *****
    noe2=sum (abs (data1-demodata ));
    nod2=length (data1) ;
    noe=noe+noe2;
    nod=nod+nod2;
    fprintf ('%d \t%e\n',iii,noe2/nod2);
end % for iii=1:nloop
% ***** Output result *****
ber = noe/nod;
fprintf ('%d \t%d\t%d\t%e \n', ebn0, noe, nod, noe/nod);
fid = fopen ('BERqam.dat', 'a' );
fprintf ( fid, '%d\t%e\t%f\t%f\n' , ebn0, noe/nod, noe, nod);
% ***** end of file*****
    figure; stem(1:41,data1(1:41),'DisplayName','data1');
    figure; stem(1:21,ich(1:21),'DisplayName','ich');
    figure; stem(1:21,qch(1:21),'DisplayName','qch');
figure;stem(1:20*IPOINT+1,ich1(1:20*IPOINT+1),'DisplayName','ich1');
    figure;
stem(1:20*IPOINT+1,qch1(1:20*IPOINT+1),'DisplayName','qch1');
figure;stem(1:40*IPOINT+1,ich2(1:40*IPOINT+1),'DisplayName','ich2');
    figure;
stem(1:40*IPOINT+1,qch2(1:40*IPOINT+1),'DisplayName','qch2');
    figure; plot(1:40*IPOINT+1,ich2(1:40*IPOINT+1),'DisplayName','ich2');
    figure;
plot(1:40*IPOINT+1,qch2(1:40*IPOINT+1),'DisplayName','qch2');
    figure;
plot(1:40*IPOINT+1,ich3(1:40*IPOINT+1),'DisplayName','ich3');
    figure;
plot(1:40*IPOINT+1,qch3(1:40*IPOINT+1),'DisplayName','qch3');
    figure;stem(1:40*IPOINT+1,ich4(1:40*IPOINT+1),'DisplayName','ich4');
    figure;
stem(1:40*IPOINT+1,qch4(1:40*IPOINT+1),'DisplayName','qch4');
    figure; stem(1:21,ich5(1:21),'DisplayName','ich5');
    figure; stem(1:21,qch5(1:21),'DisplayName','qch5');
    figure; stem(1:41,demodata(1:41),'DisplayName','demodata');
    figure; plot(ich5,qch5,'DisplayName','phaser');
    figure; stem(ich5,qch5,'DisplayName','phaser');
% *****End of file *****

```

```

% hrollfcoef.m
% Generate coefficient of Nyquist filter
function[xh]= hrollfcoef(irfn,ipoint,sr,alfs,ncc)
% alfs: rolloff coefficients
%*****
xi=zeros(1,irfn*ipoint+1);
xq=zeros(1,irfn*ipoint+1);
point=ipoint;
tr=sr;
tstp=1.0./tr./ipoint;
n=ipoint.*irfn;
mid=(n./2)+1;
sub1=4.0.*alfs.*tr; % 4*alfs*R s
    for i=1:n
        icon=i-mid;
        ym=icon;

            if icon==0.0
                xt=(1.0-alfs+4.0.*alfs/pi).*tr; % h(0)
                else
                    sub2=16.0.*alfs.*alfs.*ym.*ym./ipoint./ipoint;
                    if sub2~1.0
                        x1=sin(pi*(1.0-alfs)/ipoint*ym)./pi./(1.0-sub2)./ym./tstp;
                        x2=cos(pi*(1.0+alfs)/ipoint*ym)./pi.*sub1./(1.0-sub2);
                        xt=x1+x2; % h(t)
                    else
                        xt=alfs.*tr.*((1.0-2.0/pi).*cos...
(pi/4.0/alfs)+ (1.0+2.0/pi).*sin...
(pi/4.0/alfs))./sqrt(2.0);
                    end % if sub2 ==1.0
                    end % if icon==0
                if ncc==0 % in case of receiver
                    xh(i)=xt./ipoint./tr; % normalization
                else if ncc==1 % in the case of transmitter
                    xh(i)=xt./tr; % normalization
                else
                    error('ncc error');
                end % if ncc==0
            end % for i=1:n
        end
%*****End of file *****

```

```

% compoversamp.m
function [iout,qout]=compoversamp(idata, qdata, nsymb, sample)
%*****Variables *****
    % idata : input Ich data
    % qdata : input Qch data
    % iout : output Ich data
    % qout : output Qch data
    % nsymb : number of burst symbols
    % sample :number of oversamples
%*****
iout=zeros(1,nsymb*sample);
qout=zeros(1,nsymb*sample);
iout (1: sample: 1+sample*( nsymb-1))=idata;
qout (1: sample: 1+sample*( nsymb-1))=qdata;
%*****End of file *****

```

```

% compconv.m
% Function to perform convolution between signal and filter
function [iout, qout]=compconv (idata, qdata, filter)
%*****
% filter : filter tap coefficients
%*****
iout=conv(idata,filter);
qout=conv(qdata,filter);
%*****End of file *****

```

```

% comb.m
% Generate additive white Gaussian noise
function [iout,qout]=comb(idata, qdata,attn)
%*****Variables *****
    % iout : output Ich data
    % qout : output Qch data
    % attn : attenuation level caused by Eb/N0 or S/N
%*****
iout=randn(1,length(idata)).*attn;
qout=randn(1,length(qdata)).*attn;
iout=iout+idata(1:length(idata));
qout=qout+qdata(1:length(qdata));
%*****End of file *****

```

```

% qammod.m
% This function is used for Gray coding of 4QAM modulation
function [iout, qout] = qammod (paradata, para, nd, ml)
% ***** variables *****
% paradata : input data (para by nd matrix)
% iout : output Ich data
% qout : output Qch data
% para : Number of parallel channels
% nd : Number of data
% ml : Number of modulation levels
% (QPSK – 2 16QAM – 4)
% *****
k=sqrt(1); % The constellation power
iv=[ -1 1 ];
m2 = ml/2;
count2=0;
for ii=1 : nd
    isi = zeros (para, 1);
    isq = zeros (para, 1);
    for jj=1 : ml
        if jj <= m2
            isi = isi +2.^( m2-jj) * paradata ((1:para),...
                count2+jj);
        else
            isq = isq +2.^( ml-jj) * paradata ((1:para),...
                count2+jj);
        end
    end
    iout((1:para),ii)= iv(isi+1)./k;
    qout((1:para),ii) = iv(isq+1)./k;
    count2= count2+ml;
end
% *****end of file*****

```

```

% qamdemod.m
function [demodata] = qamdemod (idata, qdata, para, nd, ml)
% ***** variables *****
% demodata: demodulated data (para by nd matrix)
% nd : Number of data
% ml : Number of modulation levels (QPSK – 2 16QAM – 4)

```

```

k=sqrt(1);
idata= idata.*k;
qdata= qdata.*k;
demodata=zeros (para, ml*nd) ;
m2=ml/2;
count2=0;
for ii=1:nd
    a=1;
        b=1;
    i_lngth=0;
    q_lngth=0;
    for jj=1:m2
        if jj >1
            if demodata ((1:para), jj-1+count2)==1
                a=-a;
            end
        if demodata ((1:para), m2+jj-1+count2)==1
            b=-b;
        end
        i_lngth=i_lngth+i_plrty.*2.^(m2-jj+1);
        q_lngth=q_lngth+q_plrty.*2.^(m2-jj+1);
        end
    if idata ((1:para),ii) >= i_lngth
        demodata ((1:para), jj+count2)=a >=0;
        i_plrty=1;
    else
        demodata ((1:para), jj+count2)=a <=0;
        i_plrty=-1;
    end
    if qdata ((1:para) , ii) >= q_lngth
        demodata ((1:para), m2+jj+count2)=b >=0;
        q_plrty=1;
    else
        demodata ((1:para), m2+jj+count2)=b <=0;
        q_plrty=-1;
    end
end % for jj=1:m2
count2= count2+ml;
end % for ii = 1:nd
% *****end of file*****

```

## ДОДАТОК Б

Лістинг головної програми для моделювання графічної залежності BER

```

%QAM_BER
% ***** preparation part *****
sr=256000.0; % Symbol rate
ml=2; % ml : Number of modulation levels
br=sr.* ml; % Bit rate
nd = 1000
ebn0=3; % Eb/N0
IPOINT=8; % Number of oversamples
% ***** Filter initialization *****
irfn=21; % Number of taps
alfs=0.5; % Rolloff factor
[xh] = hrollfcoef (irfn,IPOINT,sr,alfs,1);
[xh2] = hrollfcoef (irfn,IPOINT,sr,alfs,0);
% ***** START CALCULATION *****
nloop=100; % Number of simulation loops
noe = 0; % Number of error data
nod = 0; % Number of transmitted data
ebn0_array=0:1:20;
ber_array=zeros(length(ebn0_array),1);
ber_array_theory=zeros(length(ebn0_array),1);
for ebn0_loop=1:length(ebn0_array)
    ebn0=ebn0_array(ebn0_loop);
    for iii=1:nloop
        % ***** Data generation *****
        data1=rand (1, nd*ml) > 0.5 ;
        % ***** 16QAM Modulation *****
        [ich qch] = qammod (data1, 1, nd, ml);
        [ich1 qch1]=compoversamp (ich, qch, length (ich), IPOINT);
        [ich2 qch2]=compconv (ich1, qch1, xh);
        % ***** Attenuation Calculation *****
        spow=sum(ich2.* ich2+ qch2.*qch2) / nd;
        attn=0.5*spow*sr/br*10.^ (-ebn0/10);
        attn=sqrt (attn);
        % ***** Add Write Gaussian Noise (AWGN) *****
        [ich3, qch3]= comb (ich2, qch2, attn) ; % add write gaussian noise
        [ich4, qch4]= compconv (ich3, qch3, xh2) ;
        sampl=irfn*IPOINT+1;
    end
end

```

```

    ich5 = ich4 (sampl : IPOINT : length (ich4)) ;
    qch5 = qch4 (sampl : IPOINT : length (ich4)) ;
% ***** MQAM Demodulation *****
[demodata]=qamdemod (ich5, qch5,1,nd,ml);
% ***** Bit Error Rate (BER) *****
    noe2=sum (abs (data1-demodata ));
    nod2=length (data1) ;
    noe=noe+noe2;
    nod=nod+nod2;
    fprintf ('%d \t%e\n',iii,noe2/nod2);
end % for iii=1:nloop
% ***** Output result *****
ber = noe/nod;
noe=0;
    nod=0;
fprintf ('%d \t%d\t%d \t%e \n', ebn0, noe, nod, noe/nod);
fid = fopen ('BERqam.dat', 'a') ;
fprintf ( fid, '%d\t%e\t%f\t%f\t\n' , ebn0, noe/nod, noe, nod);
% ***** end of file*****
ber_array(ebn0_loop)=ber;
    ber_array_theory(ebn0_loop)=0.5*erfc(sqrt(10^(ebn0_array(ebn0_loop)/10)));
%ber_array_theory1(ebn0_loop)=(3/8)*erfc(sqrt(0.4*(10^(ebn0_array(ebn0_l
oop)/10))))...
    % -(9/64)*erfc(sqrt(0.4*(10^(ebn0_array(ebn0_loop)/10))))...
    %*erfc(sqrt(0.4*(10^(ebn0_array(ebn0_loop)/10)))));
%ber_array_theory2(ebn0_loop)=(7/24)*erfc(sqrt((1/7)*(10^(ebn0_array(ebn
0_loop)/10))))...
    %-(49/384)*erfc(sqrt((1/7)*(10^(ebn0_array(ebn0_loop)/10))))...
    %erfc(sqrt((1/7)*(10^(ebn0_array(ebn0_loop)/10)))));
end
fprintf('%d\t%d\t%d\t%e\n',ebn0,noe,nod,noe/nod);
fid=fopen('BERqpsk.dat','a');
fprintf(fid,'%d\t%e\t%f\t%f\t\n',ebn0,noe/nod,noe,nod);
fclose(fid);
figure;
h=semilogy(ebn0_array,ber_array,'o',ebn0_array,ber_array_theory,'--');
%h=semilogy(ebn0_array,ber_array,'o',ebn0_array,ber_array_theory,'--');
set(h,{'DisplayName'},{'Experiment','Theory'})
legend show
% *****End of file*****

```