

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

МЕТОДИЧНІ ВКАЗІВКИ

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

Частина IV

Методичні вказівки до лабораторних робіт з дисципліни «Широкосмугові технології телекомунікацій» для студентів спеціальності 172 «Телекомунікації та радіотехніка» всіх форм навчання. Частина IV / Укл.: В.С. Кабак, Г.Ф. Вишник. – Запоріжжя: НУ «Запорізька політехніка», 2019. – 60с.

Укладачі: В.С.Кабак, доцент, к.т.н.,
Г.Ф. Вишник, зав. лаб.

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

Відповідальний.
за випуск: Г.Ф. Вишник, 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

1 ЛАБОРАТОРНА РОБОТА №7. ДОСЛІДЖЕННЯ КАНАЛУ З ЧАСТОТНО-СЕЛЕКТИВНИМИ ЗАВМИРАННЯМИ	5
1.1 Мета роботи.....	5
1.2 Завдання на підготовку до виконання лабораторної роботи	5
1.2.1 Умови виникнення частотно-селективних завмирань через часову дисперсію сигналу	5
1.2.2 Алгоритм моделювання частотно-селективних завмирань.....	11
1.3 Порядок проведення лабораторної роботи	17
1.4 Зміст звіту.....	18
1.5 Контрольні запитання	18
2 ЛАБОРАТОРНА РОБОТА №8. МОДЕЛЮВАННЯ КАНАЛУ ЗВ'ЯЗКУ З ВИКОРИСТАННЯМ КВАДРАТУРНОЇ ФАЗОВОЇ МОДУЛЯЦІЇ (QPSK)	19
2.1 Мета роботи.....	19
2.2 Завдання на підготовку до виконання лабораторної роботи	19
2.2.1 Функціональна схема I, Q - модулятора.....	19
2.2.2 Структурна схема модема QPSK.....	20
2.2.3 Реалізація програмного алгоритму для AWGN каналу	26
2.3 Порядок проведення лабораторної роботи	38
2.4 Зміст звіту.....	Ошибка! Закладка не определена. 40
2.5 Контрольні запитання	40
3 ЛАБОРАТОРНА РОБОТА №9. МОДЕЛЮВАННЯ КАНАЛУ ЗВ'ЯЗКУ В УМОВАХ РЕЛЕЇВСЬКИХ ЗАВМИРАНЬ З ВИКОРИСТАННЯМ МЕТОДУ QPSK	41
3.1 Мета роботи.....	41
3.2 Завдання на підготовку до виконання лабораторної роботи	41

3.2.1	Моделювання каналу зв'язку з використанням QPSK за наявністю частотно-неселективних або повільних завмирань	41
3.2.2	Моделювання каналу зв'язку з використанням QPSK за наявністю частотно-селективних або швидких завмирань.....	44
3.3	Порядок проведення лабораторної роботи	45
3.4	Зміст звіту.....	46
3.5	Контрольні запитання	47
	ПЕРЕЛІК ПОСИЛАНЬ	47
	Додаток А	48
	Додаток Б.....	50
	Додаток В	56

1 ЛАБОРАТОРНА РОБОТА №7 ДОСЛІДЖЕННЯ КАНАЛУ ЗВ'ЯЗКУ З ЧАСТОТНО- СЕЛЕКТИВНИМИ ЗАМИРАННЯМИ

1.1 Мета роботи

1.1.1 Ознайомитися з основними аспектами виникнення частотно-селективних завмирань через часову дисперсію сигналу.

1.1.2 За допомогою ЕОМ експериментально відпрацювати методику визначення BER в умовах частотно-селективних завмирань у каналі зв'язку з використанням модуляції BPSK.

1.2 Завдання на підготовку до виконання лабораторної роботи

1.2.1 Умови виникнення частотно-селективних завмирань через часову дисперсію сигналу

У середовищі з багатопробним розповсюдженням до антени приймача надходить не тільки пряма хвиля, а й хвилі, що надходять із затримкою, як результат відбиття, дифракції і розсіювання (рис. 1.1).

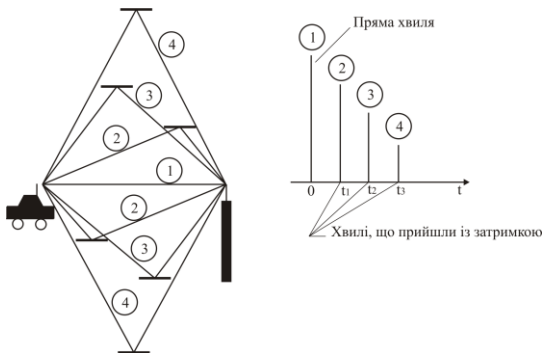


Рисунок 1.1 –Проявлення часової дисперсії сигналу

Численні перевідбиття можуть призводити до такої ситуації, коли один і той же сигнал надходить до приймача у різні моменти часу. Саме це явище й отримало назву часової дисперсії.

Наявність різних часових затримок для відбитих сигналів зумовлює виникнення міжсимвольної інтерференції (ефект накладання інформаційних бітів один на одне, яке приймач не спроможний відокремити). Часова дисперсія визначає частотно-селективний характер завмирань і може призводити до помітного погіршення якості зв'язку, а також виступає чинником, що обмежує швидкість передавання інформаційних символів у каналі.

На рис. 1.2 зображено профіль інтенсивності для такого багатопроменевого розповсюдження (залежність потужності прийнятого сигналу від затримки τ) [1].

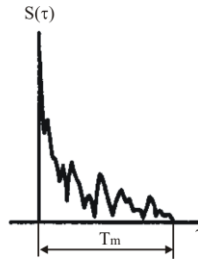


Рисунок 1.2 – Профіль інтенсивності прийнятого сигналу в умовах багатопроменевого розповсюдження

Термін часова затримка використовується для визначення надлишкової затримки розповсюдження сигналу. Практично цей термін визначає затримку даного сигналу, яка перевищує затримку надходження до приймача першого сигналу.

Для типового безпроводного каналу прийнятий сигнал звичайно складається з декількох дискретних багатопроменевих компонентів, які призводять до з'явлення ізольованих піків, які інколи називають “пальцями” або відбитими сигналами (рис.1.2). Для поодинокого переданого імпульсу час T_m між прийомом першого і останнього компонентів представляє максимальну надлишкову затримку розповсюдження, після якої потужність багатопроменевого сигналу спадає нижче визначеного порогового рівня відносно найбільш потужної компоненти.

Пороговий рівень можна обирати на 10 або на 20 дБ меншим за потужність найбільш потужного променя [1]. Для ідеальної системи (нульова надлишкова затримка) функція $S(\tau)$ складалася б з ідеального

імпульсу з ваговим коефіцієнтом, чисельне значення якого визначалося би середньою потужністю прийнятого сигналу.

У каналі із завмираннями взаємозалежність між максимальною надлишковою затримкою T_m і часом передачі символу T_s можна розглядати з позиції двох різних категорій погіршення якості передачі: частотно-селективні завмирання і частотно-неселективні або амплітудні завмирання (flat fading).

Якщо між максимальною надлишковою затримкою і тривалістю символу виконується умова:

$$T_m > T_s, \quad (1.1)$$

то це означає, що в каналі спостерігаються частотно-селективні завмирання. Очевидно, що така умова реалізується, коли прийнятий багатопроменевий компонент символу виходить за межі тривалості передачі символу і виникає ефект накладання прийнятих символів.

Іншою назвою цієї категорії погіршення якості передачі через завмирання (часову дисперсію сигналу) є введена каналом міжсимвольна інтерференція.

З іншого боку, канал можна вважати частотно-неселективним і в ньому проявляються амплітудні завмирання, якщо виконується протилежна умова:

$$T_m < T_s.$$

У цьому випадку усі прийняті багатопроменеві компоненти надходять впродовж часу передачі символу і тому відсутні спотворення за рахунок міжсимвольної інтерференції, що вноситься каналом, оскільки розширення сигналу у часі не призводить до суттєвого накладання сусідніх прийнятих символів. Але погіршення якісних характеристик передачі все ж таки має місце, оскільки деструктивне додавання компонент вектора сигналу може приводити до значного погіршення відношення сигнал/шум.

Вплив часової дисперсії на якість передачі також можна розглянути і в частотній області. На рис. 1.3 зображено кореляційну функцію рознесення частоти – це практично є Фур'є образ функції $S(\tau)$ [1, 2].

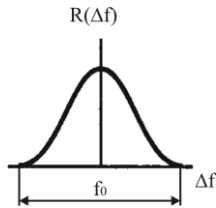


Рисунок 1.3 – Кореляційна функція рознесення частоти

Функція $R(\Delta f)$ представляє кореляційну функцію між реакціями каналу на два сигнали, як функцію різниці частот цих сигналів. Її можна розглядати як частотну передавальну функцію каналу зв'язку.

Визначивши $R(\Delta f)$ можна оцінити кореляцію між сигналами, що рознесені по частоті на величину $\Delta f = f_1 - f_2$. Практично функцію $R(\Delta f)$ можна виміряти, якщо передавати два сигнали з частотами рознесеними на величину Δf і розраховувати взаємну кореляцію спектрів отриманих сигналів з повторенням цього процесу шляхом багаторазового збільшення Δf .

Для оцінки степені кореляції вводиться поняття смуги когерентності. Смуга когерентності f_0 є статистичною мірою діапазону частот в якому канал пропускає усі спектральні компоненти приблизно з однаковим підсиленням і лінійною фазою. Тобто, у цій смузі є велика ймовірність амплітудної кореляції. Іншими словами, на усі спектральні компоненти цього діапазону канал впливає однаково, наприклад, проявляючи, або не проявляючи завмирання.

Максимальна надлишкова затримка і смуга когерентності взаємозв'язані (з точністю до постійного множника). Приблизно можна вважати, що

$$f_0 = 1/T_m. \quad (1.2)$$

Також можна відзначити, що максимальна надлишкова затримка не є оптимальним показником того, як буде функціонувати довільна система під час розповсюдження сигналу в каналі, оскільки різні канали з однаковим значенням T_m можуть мати різні профілі інтенсивності впродовж затримки. Більш коректним є застосування розкиду затримок, який найчастіше описується через середньоквадратичне значення:

$$\sigma_{\tau} = \sqrt{\tau^2 - (\bar{\tau})^2},$$

де $\bar{\tau}$ - середня надлишкова затримка, $(\bar{\tau})^2$ - квадрат середнього, τ^2 - другий момент, σ_{τ} - квадратний корінь з другого центрального моменту.

Не існує універсального співвідношення між смугою когерентності і розкидом затримок. Якщо смуга когерентності визначена, як інтервал частот в межах якого комплексна частотна передавальна функція каналу має кореляцію не менш 0,9, то смугу когерентності можна приблизно визначити як [1,2]:

$$f_0 \approx 1/(50\sigma_{\tau}).$$

Для мобільного зв'язку в якості адекватної моделі розповсюдження в умовах міста звичайно використовують сукупність розсіюючих елементів, які мають рівномірний радіальний розподіл, рівні коефіцієнти відбиття і незалежні випадкові фазові кути відбиття. Таку модель називають моделлю каналу з щільним розміщенням розсіюючих елементів. Смуга когерентності для такої моделі визначається як інтервал частот у межах якого комплексна частотна передавальна функція каналу має кореляцію не менше 0,5:

$$f_0 \cong 0,276/(\sigma_{\tau}).$$

Більш розповсюдженим наближенням для f_0 , що визначає інтервал частот для якого кореляція повинна бути не меншою 0,5, виступає наступне [1, 2]:

$$f_0 \cong 1/(5\sigma_{\tau}).$$

Розкид затримок і смуга когерентності пов'язані з характеристиками багатопроменевого розповсюдження і відрізняються для різних шляхів розповсюдження (умови міста, приміська територія, приміщення і т. ін.)

Якщо прийняти швидкість передачі символів $1/T_s$ номінально рівній швидкості передачі сигналу або смузі пропускання частот сигналу W , то згідно з (1.1, 1.2) канал називають частотно-селективним за виконанням умови:

$$f_0 < 1/T_s = W.$$

На практиці W може відрізнятись від $1/T_s$ через системну фільтрацію або вибір типу модуляції (наприклад, QPSK, MSK, розширення спектра). Частотно-селективні завмирання проявляються тоді, коли канал неоднаково впливає на різні спектральні компоненти сигналу. Деякі спектральні компоненти сигналу, що не входять до смуги когерентності, будуть підлягати різному і незалежному впливу на відміну від тих компонентів, що приходяться на смугу когерентності. Така ситуація продемонстрована на рис. 1.4 [1].

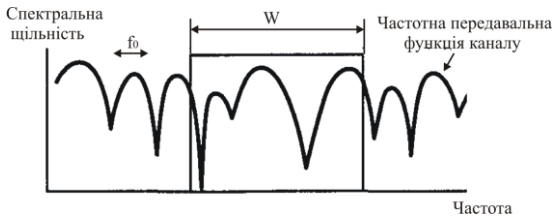


Рисунок 1.4 – Умови виникнення частотно-селективних завмирань при $W > f_0$

Частотно-неселективне завмирання або амплітудне виникає тоді, коли $f_0 > W$, тобто коли усі спектральні компоненти будуть підлягати приблизно однаковому впливу з боку каналу зв'язку. Така ситуація зображена на рис. 1.5 [1].

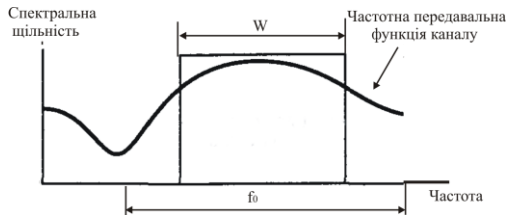


Рисунок 1.5 – Умови виникнення частотно-неселективних завмирань при $W < f_0$

В даній лабораторній роботі моделюється саме ефект частотно-селективних завмирань

1.2.2 Алгоритм моделювання частотно-селективних завмирань

Для моделювання ефекту часової дисперсії сигналу, яка призводить до міжсимвольної інтерференції, необхідно при моделюванні передбачити окрім одного шляху розповсюдження, як це було зроблено в ЛР №6, декілька шляхів розповсюдження різною довжиною. Очевидно, що довші шляхи розповсюдження призведуть до з'явлення затриманих хвиль відносно хвилі, яка надходить за найкоротшим шляхом до точки прийому (рис.1.1).

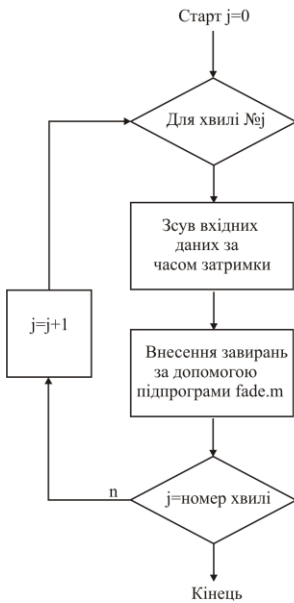
Базовою програмою для моделювання частотно-селективних завмирань у каналі зв'язку виступає програма `bpsk_fading.m`, робота якої була продемонстрована у лабораторній роботі №6.

Як відзначалося, у теоретичній частині лабораторної роботи №6, у точці прийому амплітуда розподіляється за релеївським законом, а фаза розподілена рівномірно. Прийемо, що середні потужності прямої хвилі і хвиль, що набувають затримки фіксовані. Тоді для моделювання завмирань в умовах багатопроменевого розповсюдження необхідно задати відносні рівні сигналу і відносний час затримки порівняно з прямою хвилею (рис.1.1). Блок-схема такого алгоритму зображена на рис.1.6 [3].

Цей алгоритм реалізовано у підпрограмах, текст яких приводиться у додатку Б до лабораторної роботи №6 - підпрограми `sefade.m` і `delay.m`.

Але, оскільки в ЛР №6 досліджувалися тільки повільні завмирання, які виникають через доплерівське розсіювання, то вважалось, що усі затримані хвилі сигналу відсутні і ефект часової дисперсії сигналу не проявляється. В результаті підпрограми `sefade.m`, `delay.m` практично не були задіяні, а тільки трансливали ефект доплерівського розсіювання через підпрограму `fade.m`, яка використовується як складова у підпрограмі `sefade.m`.

Для підпрограми `sefade.m` [3], вхідними даними є часовий крок, відносні рівні сигналів, відносний час затримки прямої і затриманих хвиль, послідовність модульованого сигналу, що сформована передавачем в основній смузі частот, і час моделювання одного циклу.



Вхідні параметри

Вхідні дані I каналу	:idata
Вхідні дані Q каналу	:qdata
Час затримки	:itau
Рівень потужності затриманої хвилі	:dlvl
Мінімальний часовий крок	:tstp
Час моделювання для одного циклу	:nsamp
Лічильник завмирань	:itn

Вихідні дані

Вихідні дані I каналу	:iout
Вихідні дані Q каналу	:qout

Рисунок 1.6 - Алгоритм моделювання каналу з релєївськими завмираннями

Як приклад, розглянемо випадок, коли час моделювання циклу рівний 50 мкс, а мінімальний крок у часі (змінна **tstp** у програмі) рівняється 0,5 мкс. Передбачається, що затримані хвилі мають середні потужності на 10 дБ, 20 дБ і 25 дБ менші порівняно з прямою хвилею, і час їх надходження до антени приймача відносно прямої хвилі складає 1, 1,5 і 2 мкс відповідно. Тоді вхідні змінні матимуть такий вигляд:

tstp = 0.5.*10.^(-6);

itau = [0, floor(1.*10.^(-6)/tstp), floor(1.5.*10.^(-6)/tstp),...

floor(2.0.*10.^(-6)/tstp)] = [0, 2, 3, 4];

dlvl = [0,10,20,25];

nsamp = 50.*10.^(-6)/tstp = 100;

де змінна **itau** визначає відносні затримки затриманих хвиль відносно першої, які нормовані до мінімального часового кроку **tstp**, а змінна **dlvl** визначає зменшення рівня потужностей затриманих хвиль відносно першої, рівень якої прийнятий за 0 дБ.

Визначені параметри виступають вхідними даними для симулятора (рис.1.6). В результаті такого алгоритму вхідна

послідовність затримується на час конкретної затриманої хвилі. Для кожної затриманої хвилі вводиться релієвське завмирання, яке моделюється за алгоритмом розглянутому в ЛР №6 (вираз 2.11) за допомогою підпрограми `fade.m`. Цей цикл повторюється для кожної затриманої хвилі. У підсумку вихідний сигнал отримується як результат додавання усіх хвиль, що і призводить до симуляції релієвських частотно-селективних завмирань.

Але, побудована модель має суттєвий недолік: релієвський розподіл повинен бути незалежним для кожного нового часу надходження затриманої хвилі, що запропонована модель не реалізує, оскільки до кожної хвилі застосовується один і той же математичний вираз, починаючи з часу надходження прямої хвилі. Це продемонстровано на рис. 1.7 [3].

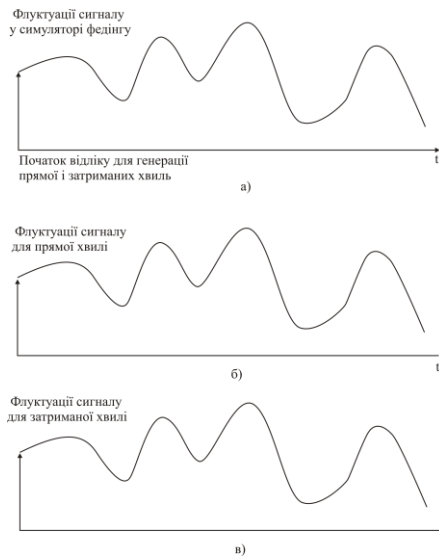


Рисунок 1.7 – Ілюстрація однакового проявлення флуктуацій сигналу для прямої та затриманої хвиль

У той же час, якщо задати різні значення початку генерації флуктуацій для прямої та затриманої хвиль, то проявлення згенерованих завмирань для прямої і затриманої хвиль буде різним

(рис.1.8). Це і означатиме, що для кожного шляху реалізується незалежний релєївський розподіл.

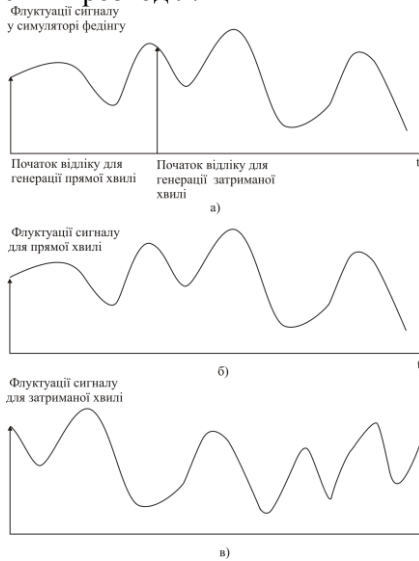


Рисунок 1.8 – Ілюстрація різного проявлення флуктуації сигналу для прямої та затриманої хвиль

Існує багато методів усунення цього недоліку. Відповідно у лабораторній роботі пропонується метод з використанням індексу завмирань або лічильника федингу [3].

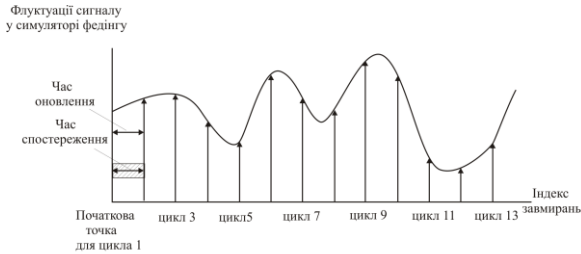
Індекс завмирань визначає час початку генерації завмирань, алгоритм яких реалізовано у програмі `fade.m`.

Розмір вектора завмирань рівняється розміру вектора, що визначає час затримки затриманої хвилі і, відповідно, вектора, який визначає відносний рівень потужності затриманої хвилі. В результаті релєївські завмирання формуються незалежно для кожного значення часу затримки (рис.1.8).

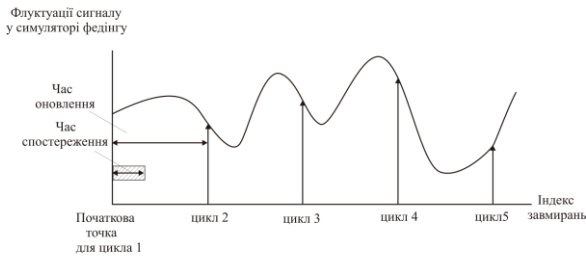
Вміст лічильника (індекс завмирань) оновлюється після кожного циклу моделювання шляхом додавання величини `itnd1`, яка визначається у відповідності до часу моделювання. Так, для розглянутого прикладу, коли час моделювання обрано рівним 50 мкс, а мінімальний крок у часі рівняється 0,5 мкс, то до вмісту лічильника після кожного циклу додається число $50/0,5=100$. Це значення

формуєтак званий час оновлення. Час оновлення може змінюватися адаптивно з метою зменшення часу моделювання.

Так, на рис.1.9а зображено змодельовані рівні сигналу, для якого проявляється ефект федингу, у випадку, коли час оновлення рівний часу спостереження.



а)



б)

Рисунок 1.9 – Співвідношення між часом спостереження і часом оновлення: (а) – час спостереження рівний часу оновлення, (б) – час спостереження менший за час оновлення

Відповідно на рис.1.9б зображено аналогічний сигнал, але для якого час оновлення більший за час спостереження [3].

У випадку, коли час оновлення рівний часу спостереження є можливість оцінити характеристики передачі практично для неперервної зміни рівня сигналу. Але, для стабілізації характеристик, що моделюються, вимагається достатньо великий час моделювання, оскільки розподіл флюктуацій сигналу буде наближатися до релеївського для дуже великої кількості циклів моделювання.

З іншого боку, коли час оновлення більший за час спостереження, то є можливість розрахувати характеристики передачі

з релеївськими завмираннями з невеликою кількістю циклів моделювання, але з меншою точністю.

В лабораторній роботі за технічним завданням передбачається проведення моделювання у випадку трьох затриманих хвиль, які затримані відносно прямої хвилі на час відповідно 1 мкс, 1,5 мкс і 2 мкс з потужностями, які на 10 дБ, 20 дБ, 25 дБ є меншими за потужність прямої хвилі.

Основною програмою для моделювання релеївських завмирань є програма `bpsk_Relay`. Зміни у програмі порівняно з програмою моделювання повільних завмирань `bpsk_fading` полягають у наступному.

Для часу надходження відповідної затриманої хвилі відносно прямої хвилі вводиться нормалізований до кроку моделювання вектор `itau`. Для швидкості передачі інформації 256 кбіт/с тривалість інформаційного біта рівняється $1/256 \cdot 10^3 = 3,9$ мкс. Порядок дискретизації у програмі прийнято рівним `IPOINT=8`. Тоді з врахуванням передискретизації мінімальний крок моделювання у часі – змінна `tstp=3,9мкс/8=0,4875` мкс. Відповідно вектор `itau` визначиться як:

`itau = [0 floor(1*10.^(-6)/tstp) floor(1.5*10.^(-6)/tstp) floor(2.0*10.^(-6)/tstp)]=[0, 2, 3, 4];`

Зменшення потужності затриманих хвиль відобразиться у векторі `dlvl: dlvl = [0,10,20,25];` .

Початкову фазу для усіх затриманих хвиль приймемо рівною нулю (вектор `th1`). Кількість хвиль для кожного шляху розповсюдження, як згадувалося, не повинна бути меншою 6. Це відобразимо у векторі `no: n0 =[6, 7, 6, 7];`

Кількість хвиль, що надійшла до приймача, відображається змінною `nowl =4` (пряма хвиля плюс три затриманих). Доплерівське розсіювання задамо таким же, як і в ЛР №6 - `fd =200`. Відповідно змінна `flat` у випадку частотно-селективних завмирань набуває нульового значення.

Остаточно блок ініціалізації параметрів федінгу у варіанті моделювання релеївських частотно-селективних завмирань приймає наступний вигляд:

```
%*****Fading initialization*****
tstp =1/sr/IPOINT;
itau=[0,2,3,4];
```

```

dlvl=[0,10,20,25];;
th1=[0.0, 0.0, 0.0, 0.0];
n0 =[6, 7, 6, 7];
itnd0=nd*IPOINT*100;
itnd1=[1000, 2000, 3000, 4000,];
nowl =4;
fd =200;
flat=0;
%*****End of fading initialization*****

```

В іншому алгоритм моделювання, задіяні підпрограми і блок виводу результатів повністю тотожний програмі `bpsk_fading`, робота якої продемонстрована у ЛР №6.

У блоці виводів результатів аналізу для графічних залежностей коефіцієнта BER від відношення E_b/N_0 передбачено, як і в ЛР №6, побудова кривих, що відповідають теоретичним виразам для AWGN каналу (позначено як '--'), релеївському порогу, який характеризує канал з частотно-неселективними (повільними) завмираннями (позначено як '*') і результати моделювання в умовах частотно-селективних завмирань відповідно до запропонованої моделі.

1.3 Порядок проведення лабораторної роботи

1.3.1 Відкрити середовище Matlab. Створити нову директорію LAB7, в яку необхідно розмістити основну програму, а також усі підпрограми, що необхідні для моделювання BPSK сигналу в умовах частотно-селективних завмирань.

1.3.2 Скопіювати файл основної програми `bpsk_Relay.m`, який знаходиться у додатку А методичних вказівок і створити новий М-файл у директорії LAB7 під тим же ім'ям.

1.3.3 Повторити процедуру створення М-файлів в директорії LAB7 для усіх підпрограм, які відповідають алгоритмам реалізації моделі модему BPSK і генерації релеївських завмирань, що використовувалися у ЛР №6 (скопіювати файли з директорії LAB6 і перенести їх до директорії LAB7):

hrollcoef.m – коефіцієнти імпульсної характеристики фільтра Найквіста (корінь квадратний);

oversamp.m – введення передискретизації до вхідної послідовності;

fade.m – генерація релеївських завмирань;

delay.m – введення часу затримки для перевідбитих хвиль;

sefade.m – формування частотно-селективних завмирань;

Після виконання вище визначених пунктів в основному вікні середовища Matlab усі 6 M-файлів повинні відобразитися у полі поточної директорії LAB7 (Current Directory).

1.3.4 У командній строчці вікна (Command Window) набрати назву основної програми `bpsk_Relay` і запустити аналіз натисканням клавіші Enter.

1.3.5 Отримати графічні залежності ймовірності помилкового прийому біта (або коефіцієнта BER) від відношення E_b/N_0 . Графічні залежності подати у напівлогарифмічному масштабі з діапазоном зміни ймовірності від 10^{-5} до 1. Діапазон зміни відношення E_b/N_0 задати від 0 до 35 дБ.

1.4 Зміст звіту

1.4.1 Структурні схеми модуляторів і демодуляторів, що досліджуються. Блок-схема алгоритму програми.

1.4.2 Графічні залежності $BER=f(E_b/N_0)$ за результатами моделювання та теоретичними виразами.

1.4.3 Лістинги програм

1.4.4 Висновки за результатами роботи.

1.5 Контрольні запитання

1. Поняття часової дисперсії.
2. Типовий профіль інтенсивності прийнятого сигналу в умовах багатопроменевого сигналу.
3. Визначення часу максимальної надлишкової затримки.
4. Визначення смуги когерентності через максимальну надлишкову затримку.
5. Визначення смуги когерентності через середньоквадратичне значення максимальної затримки.
6. Умови виникнення частотно-селективних завмирань у часовій області.
7. Умови виникнення частотно-селективних завмирань у частотній області.

8. Умови виникнення частотно-неселективних завмирань у частотній області.
9. Умови виникнення частотно-неселективних завмирань у часовій області.
10. Методи і засоби боротьби з релієвськими завмираннями.

2 ЛАБОРАТОРНА РОБОТА №8 МОДЕЛЮВАННЯ КАНАЛУ ЗВ'ЯЗКУ З ВИКОРИСТАННЯМ КВАДРАТУРНОЇ ФАЗОВОЇ МОДУЛЯЦІЇ (QPSK)

2.1 Мета роботи

2.1.1 Ознайомитися з принципами побудови модуляторів і демодуляторів для методів квадратурної фазової модуляції КФМ-4 (QPSK).

2.1.2 За допомогою ЕОМ експериментально відпрацювати методику формування модулюючого сигналу, каналних і радіочастотних символів QPSK модуляторів, розглянути особливості побудови демодуляторів.

2.1.3 Навчитися будувати діаграму фазових переходів. Отримати фазові сузір'я для різних значень E_b/N_0 .

2.2 Завдання на підготовку до виконання лабораторної роботи

2.2.1 Функціональна схема I, Q - модулятора

На відміну від двійкової фазової маніпуляції, де один каналний радіосимвол переносить один інформаційний біт, квадратурна фазова маніпуляція дозволяє реалізувати передачу одним каналним радіосимволом двох інформаційних бітів.

Для реалізації методу QPSK звичайно використовують схему I, Q модулятора (рис.2. 1) [4].

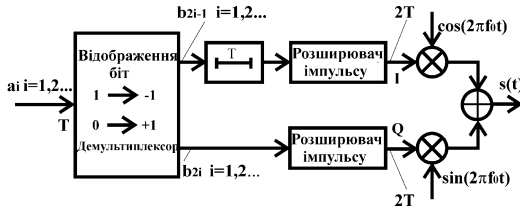


Рисунок 2.1 – Функціональна схема модулятора КФМ-4 (QPSK)

Для визначення властивостей модульованих сигналів досить часто використовують діаграму фазових переходів, яка відтворює графічне зображення траєкторій переміщення сигнальних точок у сигнальному сузір'ї під час переходу від одного канального символу, що передається, до іншого.

Введення символу I для дійсної частини комплексного символу і символу Q для уявної частини дозволяє представляти як сигнальне сузір'я, так і діаграму фазових переходів у координатах I, Q . Саме таке зображення фазових переходів у координатах I, Q отримало у літературі назву фазового сузір'я (англійською – phase constellation). Для сигналу QPSK діаграма фазових переходів зображена на рис. 2. 2. [4, 5].

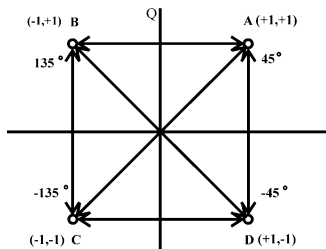


Рисунок 2. 2 – Фазове сузір'я (діаграма фазових переходів) для сигналу QPSK

2.2 2 Структурна схема модему QPSK

На підставі запропонованої функціональної схеми модулятора QPSK (рис.2.1) визначимо структурні схеми передавального і приймального трактів.

З функціональної схеми модулятора виходить, що радіосигнал з QPSK модуляцією формується шляхом додавання радіосигналів, які формуються у двох незалежних квадратурних I, Q каналах.

Програмне формування модулюючих каналних I, Q символів для проведення моделювання пропонується реалізувати на підставі послідовно-паралельного перетворювача, алгоритм роботи якого продемонстровано на рис.2.3.

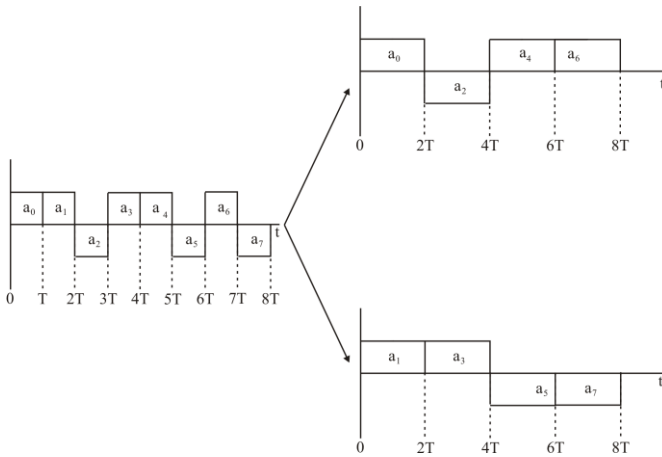


Рисунок 2. 3 – Алгоритм формування каналних символів для методу QPSK

Відповідно, після формування каналних I, Q символів за таким алгоритмом, в кожному з каналів здійснюється класична BPSK модуляція і, відповідно, обробка сигналів у кожному з каналів моделі проходить повністю тотожно до методу BPSK.

Для забезпечення передачі інформації в каналі зв'язку без міжсимвольної інтерференції необхідно для модулюючої послідовності застосувати формуючий фільтр Найквіста.

Тому після генерації вхідних цифрових даних необхідно реалізувати формування каналних символів для кожного з каналів модулятора QPSK відповідно до рис.2.3, перетворити їх до формату NRZ і подальшу обробку модулюючого сигналу провести у формуючому фільтрі з передавальною функцією $G_r(f) = \sqrt{G_N(f)} = \sqrt{RC}$

(повністю тотожному фільтру, якого було запропоновано в ЛР №5, ЛР №6, ЛР №7).

Формування каналних символів відбувається практично у програмному вигляді з використанням класичних методів цифрової обробки сигналів і самі символи подаються у дискретному вигляді – тобто у вигляді цифрових вибірок.

Як і для класичного методу BPSK, перенесення сигналу на радіочастоту відбувається аналоговими засобами, зокрема, як на схемі модулятора (рис.2.3) це реалізується шляхом перемноження модулюючих символів на коливання відповідної високочастотної квадратури. Тому необхідно перетворити відфільтрований сигнал до аналогової форми за допомогою цифро-аналогового перетворювача (ЦАП).

Результуючий сигнал після перемножувачів і є вихідним сигналом кожного з квадратурних каналів, які подаються до суматора.

Відповідна структурна схема передавального тракту з використанням методу QPSK зображена на рис. 2.4.

Структурна схема приймального тракту для методу QPSK зображена на рис.2.5. 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. Аналогічно, як і для методу BPSK, використовується схема когерентного прийому.

Для цього на приймальному боці формується поновлене несуче коливання. Відповідно для I-каналу таким коливанням виступає $\cos\omega_0 t$, а для квадратурного каналу $\sin\omega_0 t$.

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

Подальша обробка сигналу на практиці знову відбувається із застосуванням DSPH (digital signal processor hardware) апаратури, яка передбачає застосування цифрових методів обробки сигналів.

Відповідно сигнал знову необхідно подати у вигляді цифрових вибірок, для чого використовується аналогово-цифровий

перетворювач (АЦП). Аналогічно, як і для методу BPSK, у кожному каналі демодулятора застосовується такий же формуючий цифровий фільтр \sqrt{RC} , як і на передавальному боці

Для відфільтрованого передискретизованого сигналу необхідно обрати точку синхронізації і провести операцію зворотну передискретизації. Останнім кроком є застосування порогового пристрою, який приймає рішення за простим алгоритмом: якщо значення вибірки більше 0, то приймається рішення про прийом логічної “одиниці”, якщо ж значення вибірки менше 0, то приймається рішення про прийом логічного “нуля”.

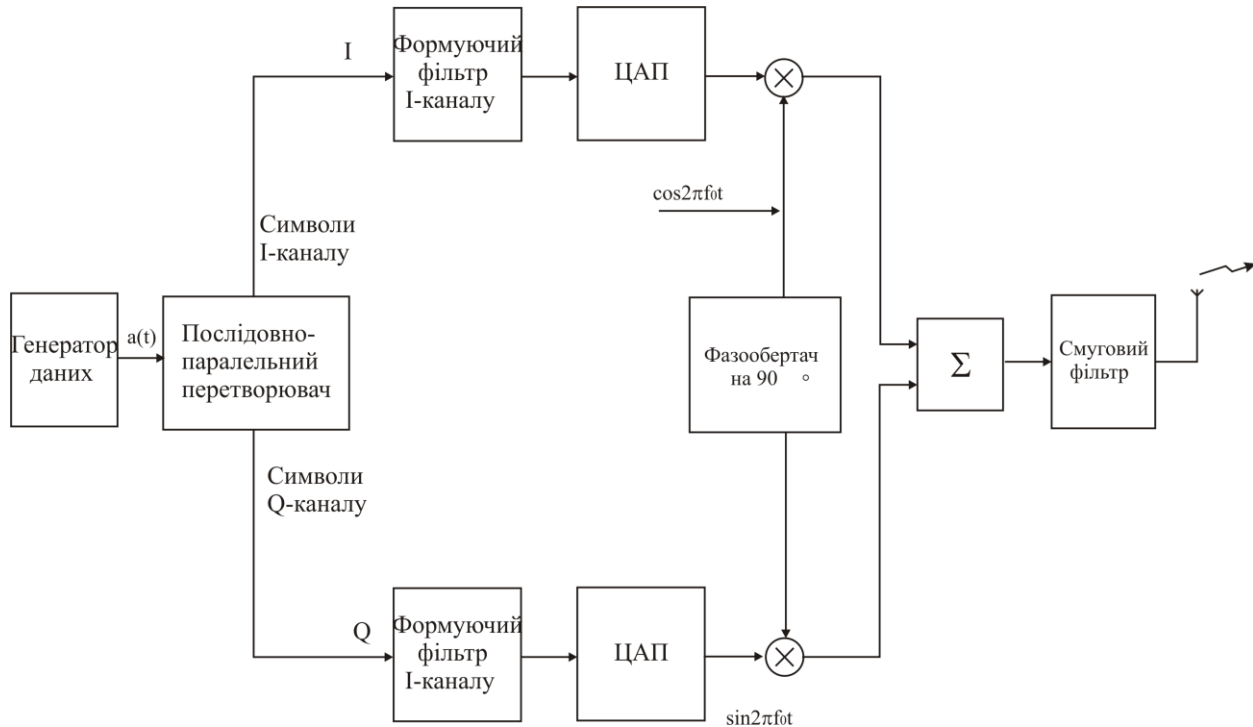


Рисунок 2. 4 – Структурна схема передавального тракту каналу зв'язку з використанням методу QPSK

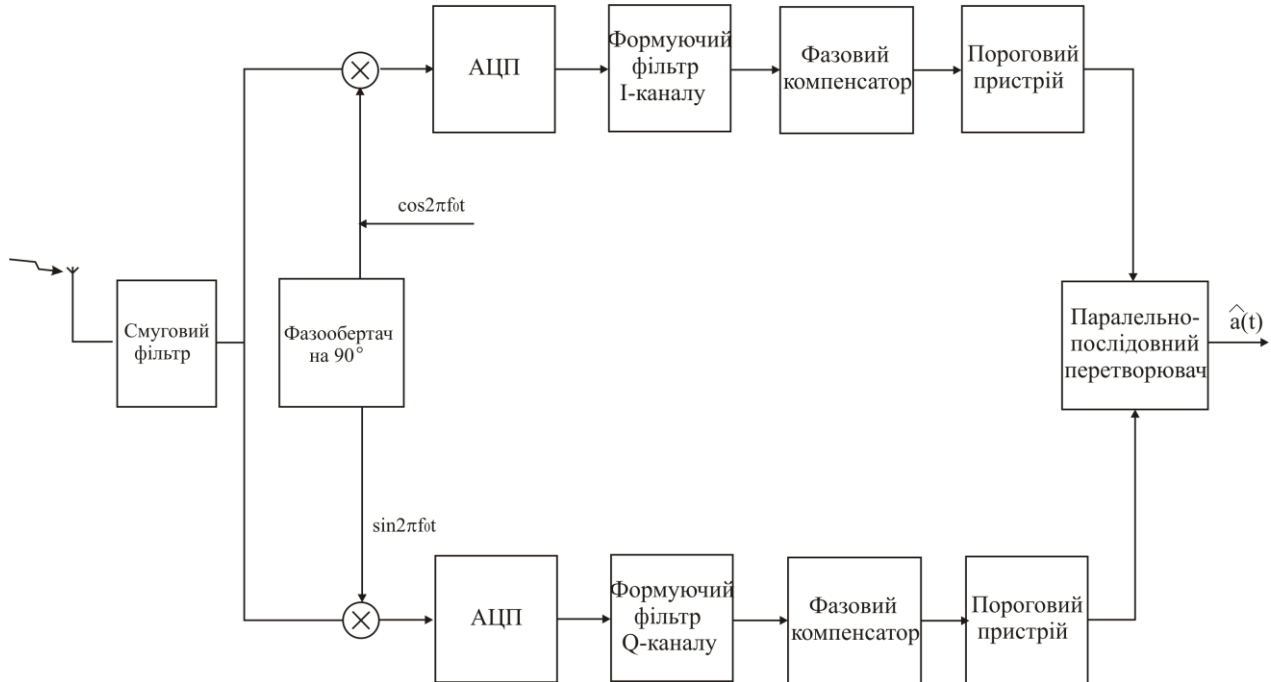


Рисунок 2. 5 – Структурна схема приймального тракту каналу зв'язку з використанням методу QPSK

Як і для методу BPSK питання поновлення несучого коливання, а також питання тактової синхронізації не розглядаються, а вирішуються основні методи реалізації модему в основній смузі частот.

2.2.3 Реалізація програмного алгоритму для AWGN каналу з QPSK модуляцією

Блок-схема каналу зв'язку, яка пропонується для моделювання, зображена на рис. 2. 6.

Структурна схема кожного з квадратурних каналів I,Q модулятора і демодулятора моделі подібна до розглянутої у ЛР №5 схеми каналу зв'язку з BPSK модуляцією. Тому за основу програми береться вже розроблений набір підпрограм для BPSK модуляції.

Швидкість передавання символів також задана рівною 256000 біт/с, або 256 кбіт/с, кількість бітів ml , що переносяться за один радіосимвол, на відміну від методу BPSK прийнята рівною $ml=2$, відповідно бітова швидкість br визначається як результат перемноження символної швидкості на параметр ml , що у випадку QPSK означає, що бітова швидкість удвічі більша за символну швидкість.

Кількість символів, що підлягають аналізу для однієї реалізації випадкового процесу прийнята рівною $nd=1000$, початкове значення параметру $E_b/N_0=3$ дБ.

В термінах алгоритмічної мови програмування середовища Matlab блок введення початкових даних для QPSK модуляції набуває наступного вигляду:

```
% qpsk.m
% Simulation program to realize QPSK transmission system
%*****Preparation part*****
sr=256000.0; % Symbol rate
ml=2; % ml : Number of modulation levels
br=sr.*ml; % bit rate
nd=1000; % Number of symbols
ebn0=3; % Eb/N0
IPOINT=8; % Number of oversamples
```

Наступним блоком програми є завдання параметрів формуючого фільтра Найквіста.

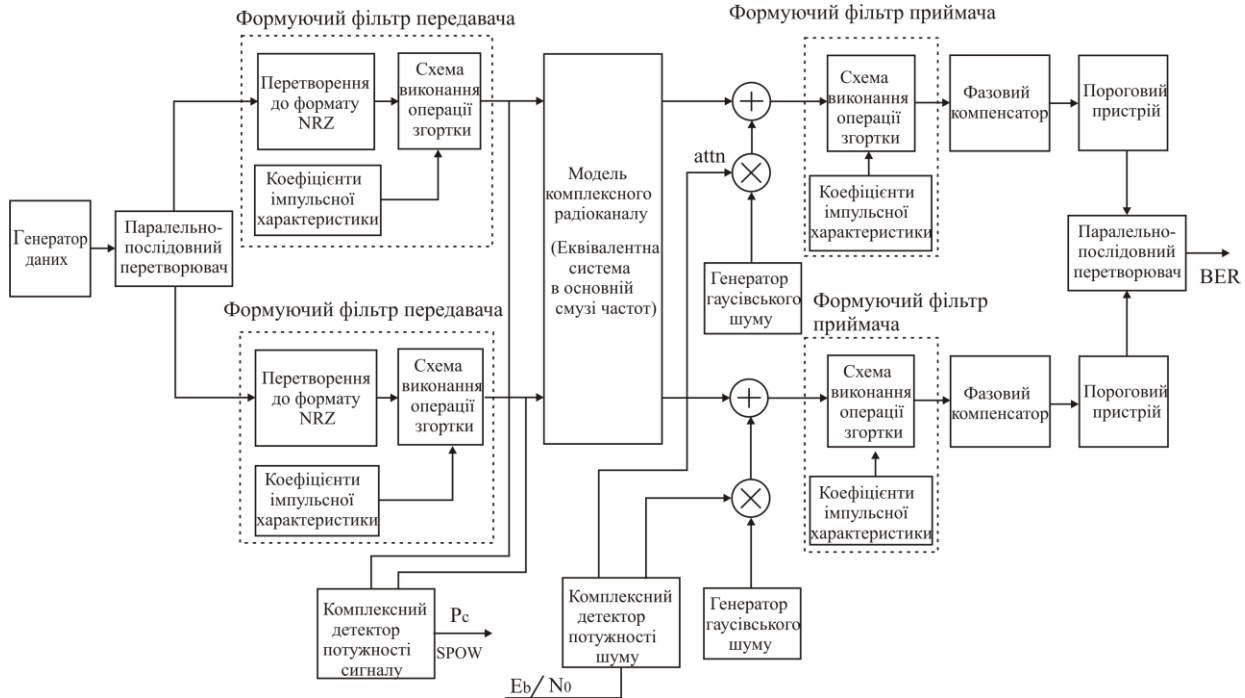


Рисунок 2.6 – Блок-схема каналу зв'язку з використанням QPSK для моделювання в основній смузі частот

Оскільки символна швидкість обрана такою ж, як і для методу BPSK, то коефіцієнти імпульсної характеристики фільтра \sqrt{RC} розраховуються за допомогою підпрограми **hrollfcoef**(irfn, IPOINT, sr, alfs, 1/0), що була задіяна у лабораторних роботах №5, №6, №7. Відповідно блок ініціювання параметрів фільтра залишається таким же, як і у головних програмах моделювання каналу з використанням методу BPSK.

Подалі починається основне тіло програми.

Для того, щоб адекватно визначити імовірнісні 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.

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

```

Тіло циклу починається з генерації вхідних даних, яка відбувається подібно до методу BPSK завданням вектора **data1** розмірністю `1*nd*ml`:

```

%*****Data generation*****
data1=rand(1,nd*ml)>0.5    % rand: built in function.

```

Наступним кроком відповідно до блок-схеми (рис.2.6) є формування послідовності парних та непарних символів I- каналу - вектор [ich] та Q-каналу - вектор [qch] згідно з алгоритмом, що наведено на рис.2.3. Цей алгоритм послідовно-паралельного перетворення реалізується за допомогою підпрограми `qpskmod.m` [3]:

```

% qpskmod.m
% Function to perform QPSK modulation
function [iout,qout]=qpskmod(paradata, para, nd, ml)

```

Змінні підпрограми `qpskmod.m` 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.

Відповідно в підпрограмі спочатку вхідні дані перетворюються на послідовність у форматі NRZ без постійної складової, а потім формуються дві підпослідовності (два вектори) **isi** та **isq** парних і непарних символів.

Оскільки цю підпрограму планується використовувати як для QPSK модуляції, де тривалість символу $T_s=2T_6$, так і для багатопозиційної QAM, то в підпрограмі організовано додатковий цикл (змінна циклу - **ii**), який передбачає формування груп з 4,5,6 бітів для методів 16-QAM, 32-QAM і 64-QAM. Безпосередньо для методу QPSK цей цикл виконується тільки один раз і формує підпослідовності парних **isi** і непарних **isq** символів з вхідної послідовності. Повий лістинг підпрограми наведено у додатку Б.

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

На рис.2.7 зображено послідовність згенерованих випадкових даних (**data1**) у вигляді цифрових відліків, а на рис. 2.8, рис.2.9 сформовані в результаті роботи програми **qpskmod.m** підпослідовності парних (**ich**) та непарних (**qch**) відліків у форматі NRZ для I та Q -каналів відповідно.

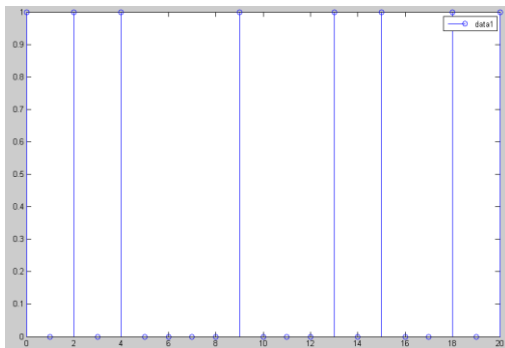


Рисунок 2.7 – Згенерована послідовність цифрових відліків

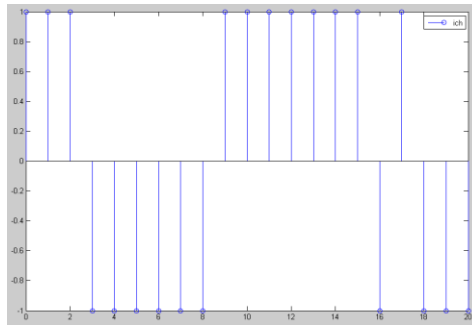


Рисунок 2. 8 –Підпоследовність парних цифрових відліків I-каналу

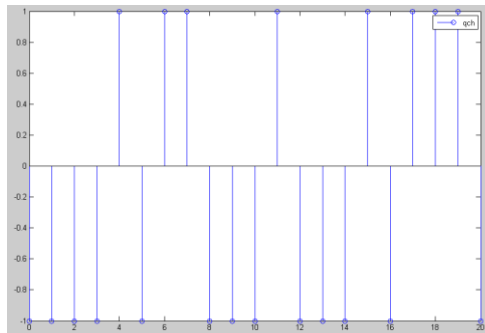


Рисунок 2. 9 –Підпоследовність непарних цифрових відліків Q-каналу

Далі необхідно здійснити операцію передискретизації цифрових відліків в кожному каналі I,Q – модулятора з наступною фільтрацією у формуючому фільтрі Найквіста. Це виконується за допомогою підпрограм **compoversamp.m**, **compconv.m**.

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

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

Підпрограма **compoversamp.m** практично є розширенням підпрограми **oversamp.m**, яка реалізовувала операцію

передискретизації для методу BPSK. Відмінність полягає у тому, що для BPSK формувався тільки один вектор з додатковими нульовими відліками згідно з порядком передискретизації, а для QPSK формується два таких вектори у кожному каналі **ich1** і **qch1**, які формують комплексну послідовність передискретизованих каналних символів **[ich1,qch1]** з комплексної послідовності **[ich,qch]**.

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

Програма **compconv.m** також є розширенням програми **conv.m**, яка реалізувала операцію згортки вхідних даних з коефіцієнтами імпульсної характеристики формуючого фільтра Найквіста .

Відмінність також полягає у тому, що для QPSK необхідно сформувати два вихідних вектори **ich2** і **qch2**, для кожного з яких необхідно застосувати операцію згортки.

Результатом роботи програми виступає двомірний вектор **[ich2,qch2]**, який формує комплексну послідовність відфільтрованих каналних символів **[ich2,qch2]** з комплексної вхідної послідовності **[ich1,qch1]**.

Змінними підпрограми виступають вектори **ich1** і **qch1** і коефіцієнти імпульсної характеристики формуючого фільтра **xh**.

Відповідні епюри каналних символів, після передискретизації для I (**ich1**) та Q-каналів (**qch1**) зображено на рис.2.10, рис.2.11.

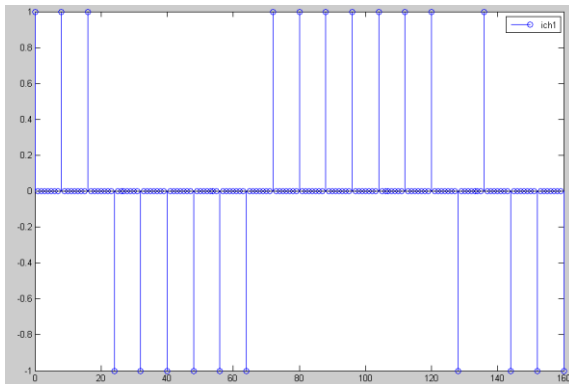


Рисунок 2.10 – Підпослідовність парних цифрових відліків I-каналу після передискретизації

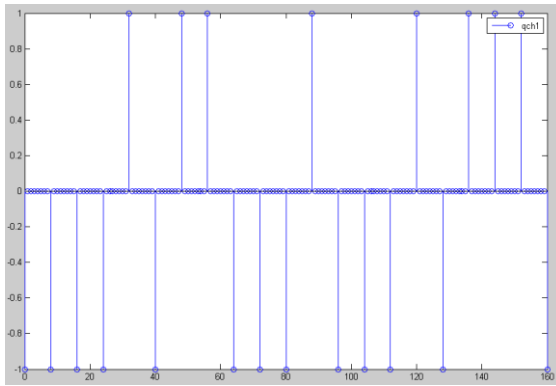


Рисунок 2.11 – Підпоследовність непарних цифрових відліків Q-каналу після передискретизації

Під час аналізу наведених епоюр необхідно звернути увагу на зміну масштабу по осі X, що пояснюється введенням додаткових нулів на кожен символ - 20 відлікам відповідає $20 \cdot 8 = 160$ вибірок.

Сформовані сигнали на виході формуючого фільтра \sqrt{RC} в кожному з каналів, як результат роботи підпрограми **comprconv.m**, зображено на рис.2.12,2.13.

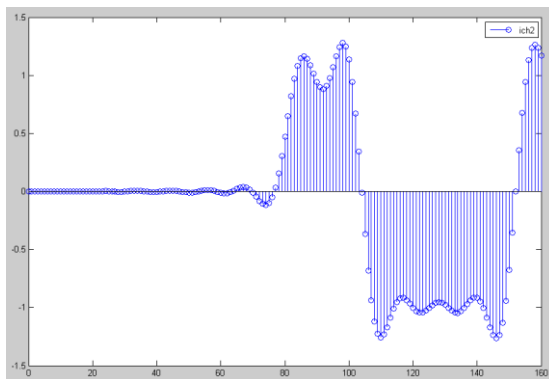


Рисунок 2.12 – Відфільтрована підпоследовність парних цифрових відліків I-каналу

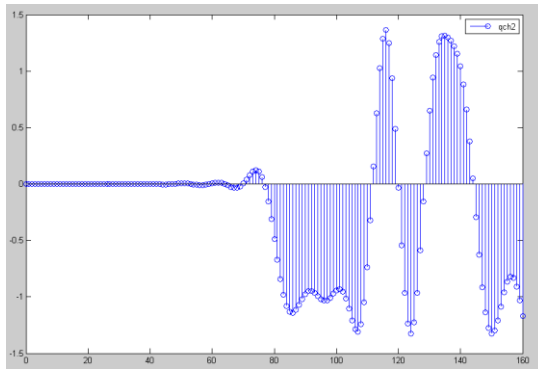


Рисунок 2.13 –Відфільтрована підпоследовність непарних цифрових відліків Q-каналу

Для здійснення безпосередньо BPSK модуляції в кожному каналі необхідно цифровий сигнал перетворити до аналогової форми за допомогою ЦАП. Результуючі послідовності каналних символів I та Q-каналів після ЦАП, які безпосередньо модулюють квадратурні несучі зображено на рис.2.14, 2.15.

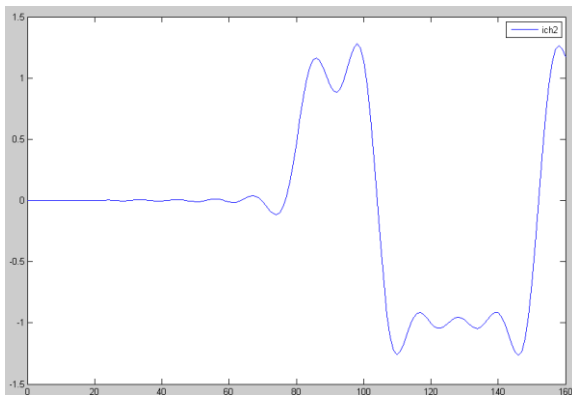


Рисунок 2.14 –Модулюючі каналні символи I-каналу

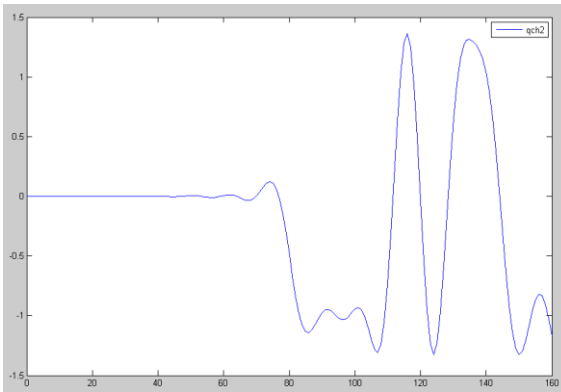


Рисунок 2.15 –Модуючі каналні символи Q-каналу

Моделювання обробки сигналів на приймальному боці починається з визначення відношення сигнал/шум для прийнятого сигналу у припущенні, що розглядається AWGN канал. Це здійснюється подібно до методу BPSK, але потужність, що приходить на один біт **SPOW**, визначається як середньоквадратична сума усіх символів I і Q – каналів, що поділена на кількість інформаційних біт:

```
%***** Attenuation Calculation*****
```

```
spow=sum(ich2.*ich2+qch2.*qch2)/nd;
```

```
% sum: built in function
```

```
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**. Ця підпрограма також є розширенням алгоритму запропонованого для методу BPSK, де відбувалася генерація гаусівського шуму для одного каналу, а коефіцієнт **attn** визначав абсолютну потужність шуму. В підпрограмі **comb.m** подібна процедура застосовується для обох каналів демодулятора з наступним додаванням визначених шумів до корисного сигналу в обох каналах демодулятора:

```
% comb.m
```

```
% Generate additive white Gaussian noise
```

```

function [iout,qout]=comb(idata, qdata,attn)
    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 *****

```

В результаті комплексний сигнал на вході демодулятора (в основній смузі частот) з урахуванням впливу шумів визначиться як двовірний вектор **[ich3,qch3]**:

```
[ich3,qch3]=comb(ich2,qch2,attn); % add white Gaussian noise
```

Епюри коливань, які формуються на входах I, Q - демодулятора в результаті додавання гаусівського шуму при значенні $E_b/N_0=10$ дБ наведено на рис.2.16, рис.2.17 (порівняти з рис. 2.14, 2.15).

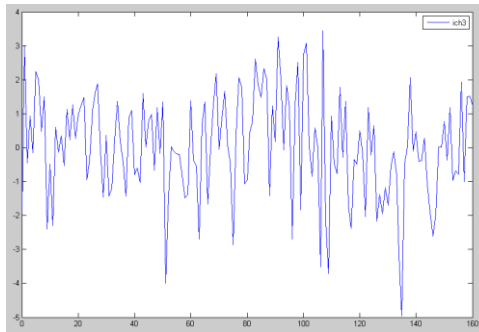


Рисунок 2.16 – Сигнал основної смуги частот на вході I-каналу демодулятора

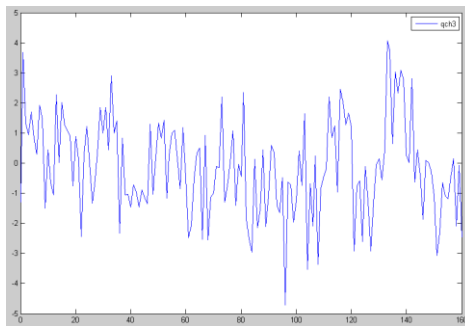


Рисунок 2.17 – Сигнал основної смуги частот на вході Q-каналу демодулятора

Подальша обробка символів в кожному з каналів демодулятора здійснюється подібно до методу BPSK – реалізується фільтрація у формуючому фільтрі з формуванням вектора **[ich4,qch4]** за допомогою підпрограми **compconv.m**, здійснюється вибір точки синхронізації і усунення передискретизації відліків в I і Q каналах:

```
[ich4,qch4]=compconv(ich3,qch3,xh2);
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** з урахуванням впливу шумів.

Остаточню, результат демодуляції сигналів демодулятора у вигляді однополярної послідовності поновлених відліків формується за допомогою підпрограми **qpskdemod.m**:

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

Підпрограма **qpskdemod.m** практично реалізує алгоритм роботи порогових пристроїв в I, Q каналах демодулятора і реалізує функцію паралельно-послідовного перетворювача:

```
% qpskdemod.m
% Function to perform QPSK demodulation
function [demodata]=qpskdemod(idata, qdata, para, nd, ml)
%*****
demodata==zeros(para,ml*nd);
demodata==((1:para),(1:ml:ml*nd-1))=idata((1:para),(1:nd))>=0;
demodata==((1:para),(2:ml:ml*nd-1))=qdata((1:para),(1:nd))>=0;
%*****End of file*****
```

В результаті роботи запропонованого програмного комплексу на виході модему QPSK сигналів формується демодульована послідовність для заданого відношення E_b/N_0 .

Окрім, власне, моделювання процесів формування і обробки сигналів в модуляторі і демодуляторі методу QPSK, розроблений комплекс 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 відкладене значення демодульованих двополярних відліків I-каналу, а по осі ординат відповідно значення демодульованих відліків Q-каналу (до застосування порогових

пристроїв, які реалізуються у підпрограмі qpskdemod.m). Це можливо за рахунок введення до блоку виводу результатів аналізу оператора:

figure; plot(ich5,qch5,'DisplayName','phaser');

Відзначимо, що фазове сузір'я методу QPSK, яке було зображено на рис.2.2, відповідає умовам ідеального каналу без шумів, тобто у випадку, коли послідовності на вході демодулятора приймача повністю повторювали б відповідні сигнали на виході передавача (рис.2.14, рис.2.15). Вплив гаусівського шуму на вході приймача суттєво спотворює форму прийнятого сигналу (рис.2.16, рис.2.17), що в результаті призведе і до спотворень форми фазової діаграми. Так, для умов заданих в лабораторній роботі, форма отриманої фазової діаграми при відношенні $E_b/N_0=10$ дБ зображена на рис.2.18.

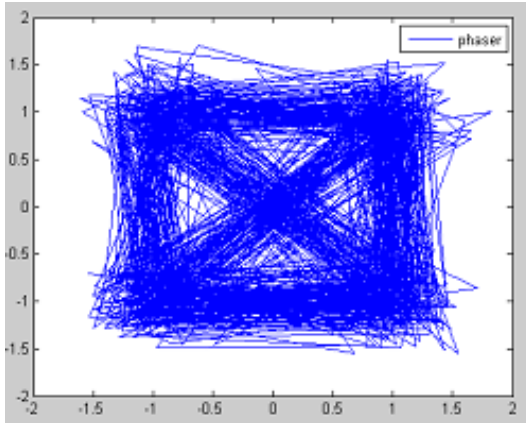


Рисунок 2.18 – Діаграма фазових переходів при значенні $E_b/N_0=10$ дБ

Очевидно, що діаграма фазових переходів за наявності гаусівських шумів у каналі зв'язку суттєво відрізняється від теоретичної, яку наведено на рис.2.2, особливо для малих значень E_b/N_0 . Як наслідок, можна очікувати, що значення BER за таких умов буде достатньо великим.

Алгоритм визначення коефіцієнта BER повністю тотожний методу BPSK. Ймовірність помилкового прийому одного біта розраховується як відношення кількості помилково прийнятих бітів до загальної кількості переданих бітів.

Теоретичний вираз для ймовірності помилкового прийому одного біта або коефіцієнта бітової помилки (BER) для AWGN каналу при умові відносного кодування даних і реалізації когерентного детектування має такий вигляд: [3,5]:

$$\text{BER}_{\text{QPSK,AWGN}} = \frac{1}{2} \text{erfc}(\sqrt{E_b/N_0}). \quad (2.1)$$

Програма `qpsk.m` передбачає вивід результатів, щодо ймовірності помилкового прийому у вигляді масиву чисел для одного конкретного значення E_b/N_0 для заданої кількості реалізацій випадкового процесу.

Отримання графічної залежності $\text{BER}=F(E_b/N_0)$ реалізовано у програмі `qpsk_ber.m`. Практично ця програма в головному повторює `qpsk.m`, але, з метою формування графічної залежності $\text{BER}=F(E_b/N_0)$ перед основним циклом (оператор **for iii=1:nloop**) створюється додатковий цикл за параметром E_b/N_0 .

Організація циклу і блок виводу результатів повністю тотожні аналогічним блокам, які застосовувалися в ЛР №5, ЛР №6, ЛР №7

Блок виводу результатів передбачає вивід графічних залежностей, що отримані в результаті моделювання за запропонованим алгоритмом реалізації модему (позначено як 'o') і теоретичним виразом (2.1) (позначено як '-' - 'o').

2.3 Порядок проведення лабораторної роботи

2.3.1 Відкрити середовище Matlab. Створити нову директорію (папку) LAB8 в яку необхідно розмістити основну програму, а також усі підпрограми, що необхідні для моделювання QPSK сигналу для AWGN каналу.

2.3.2 Скопіювати файл основної програми `qpsk.m`, який знаходиться у додатку Б методичних вказівок і створити новий М-файл у директорії LAB8 під тим же ім'ям.

2.3.3 Повторити процедуру створення М-файлів в директорії LAB8 для усіх підпрограм, які відповідають алгоритмам реалізації моделі модему QPSK (лістинги програм наведені у додатку Б методичних вказівок):

hrollcoef.m – коефіцієнти імпульсної характеристики фільтра Найквіста (ЛР №5, ЛР №6, ЛР №7);

qpskmod.m – формування каналних символів для методу QPSK;

compoversamp.m – передискретизація в обох каналах I, Q модулятора;

compconv.m – операція згортки в обох каналах I, Q модулятора;

qpskdemod.m – алгоритм роботи порогових пристроїв;

comb.m – додавання гаусівського шуму до обох каналів I, Q демодулятора;

qpsk_ber – програма виводу графічної залежності BER.

Після виконання вище визначених пунктів в основному вікні Matlab усі 8 файлів повинні відобразитися у полі поточної директорії (Current Directory).

2.3.4 У командній строчці командного вікна (Command Window) набрати назву основної програми qpsk і запустити аналіз натисканням клавіші Enter.

Якщо синтаксичні помилки у програмах відсутні, то на дисплей у командному вікні повинні відобразитися значення BER для кожної реалізації випадкової послідовності відповідно до заданого циклу **nloop**.

2.3.5 Зафіксувати епюри отриманих сигналів і діаграму фазових переходів. Поставити у відповідність отримані епюри і входи/виходи відповідних блоків моделі каналу зв'язку.

2.3.6 Відкрити файл qpsk.m. Змінити значення E_b/N_0 у блоці *****PreparationPart***** на значення **$E_b/N_0=10\text{дБ}$** , зберегти файл. Запустити аналіз. Зафіксувати нове фазове сузір'я.

2.3.7 Повторити процедуру моделювання для значень $E_b/N_0=20\text{дБ}$ і $E_b/N_0=35\text{дБ}$. Зафіксувати отримані фазові сузір'я.

2.3.8 Отримати графічну залежність ймовірності помилкового прийому біта (або коефіцієнта BER) від відношення E_b/N_0 . Для цього у командній строчці набрати назву програми qpsk_ber і запустити аналіз.

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

2.3.9 Порівняти залежності BER для AWGN каналу при використанні методів BPSK і QPSK.

2.4 Зміст звіту

2.4.1 Структурні схеми модуляторів і демодуляторів, що досліджуються. Блок-схема алгоритму програми.

2.4.2 Епюри отриманих сигналів на входах/виходах функціональних вузлів блок-схеми каналу зв'язку.

2.4.3 Діаграми фазових переходів при різних значеннях E_b/N_0 .

2.4.4 Графічні залежності $BER=f(E_b/N_0)$ за результатами моделювання та теоретичними виразами.

2.4.5 Лістинги програм

2.4.6 Висновки за результатами роботи.

2.5 Контрольні запитання

1. Функціональна схема модулятора QPSK сигналу.
2. Функціональна схема демодулятора QPSK сигналу.
3. Формування каналних символів для методу QPSK.
4. Алгоритм введення відношення E_b/N_0 до моделі каналу зв'язку.
5. Призначення формуючого фільтра Найквіста в модемі QPSK.
6. Обґрунтувати необхідність введення передискретизації в модемах цифрових видів модуляції.
7. Як змінюється фазове сузір'я під час зміни відношення E_b/N_0 .
8. Яким виразом описується залежність ймовірності помилкового прийому біта для AWGN каналу при використанні методу QPSK.
9. Провести порівняльний аналіз залежності BER від відношення E_b/N_0 для AWGN каналу при використанні методів QPSK і BPSK.
10. Обґрунтувати збіг залежностей BER від відношення E_b/N_0 для AWGN каналу при використанні методів QPSK і BPSK.

3 ЛАБОРАТОРНА РОБОТА №9 ДОСЛІДЖЕННЯ КАНАЛУ ЗВ'ЯЗКУ В УМОВАХ РЕЛЕЇВСЬКИХ ЗАВМИРАНЬ ПРИ ВИКОРИСТАННІ МЕТОДУ QPSK

3.1 Мета роботи

За допомогою ЕОМ експериментально відпрацювати методику визначення BER в умовах релеївських завмирань у каналі зв'язку з використанням модуляції QPSK.

3.2 Завдання на підготовку до виконання лабораторної роботи

3.2.1 Моделювання каналу зв'язку з використанням QPSK за наявністю частотно-неселективних або повільних завмирань

Основною програмою для моделювання каналу зв'язку з використанням QPSK за умови наявності частотно-неселективних завмирань у каналі виступає програма `qpsk_fading.m`, яка практично є розширенням програми `qpsk.m`, що розглянута у попередній ЛР №8.

Зміни у програмі полягають у наступному.

Після завдання вхідних даних для моделювання і ініціалізації параметрів формуючого фільтра Найквіста включається блок ініціалізації параметрів федингу.

Аналогічно, як і для BPSK, за перший варіант моделювання приймається гіпотетичний канал, для якого частотно-селективні та швидкі завмирання відсутні, а проявляються тільки повільні завмирання, які характеризуються кривою релеївського порогу. Це означає, що розглядається тільки один шлях розповсюдження хвилі (за довжиною) і затримані хвилі відсутні (відсутній ефект міжсимвольних спотворень). Відповідно усі змінні в програмі, що відносяться до затриманих хвиль, покладаються рівними 0.

Ширина доплерівського розсіювання або швидкість завмирань рівняється як і для BPSK $\Delta f_d = 200$ Гц. Оскільки швидкість передачі символів залишається рівною $R = 256$ кбіт/с, то ефект доплерівського розсіювання проявляється як частотно-неселективні повільні

завмирання. Тобто, змінну **flat** покладемо рівній одиниці і блок ініціалізації параметрів федингу приймає вид повністю тотожний методу BPSK:

```
%*****Fading initialization*****
tstp =1/sr/IPOINT;
itau=0;
dlvl =0;
th1=[0.0];
n0 =6;
itnd0=nd*IPOINT*100;
itnd1=[1000];
nowl =1;
fd =200;
flat=1;
%*****End of fading initialization*****
```

Як було визначено у ЛР №6, введення доплерівського розсіювання і часової дисперсії сигналу реалізується підпрограмою **sefade.m**. Вхідними даними для програми **sefade.m** виступають вихідний сигнал I,Q модулятора (ich2, qch2 у програмі і параметри, які введені у блоці ініціалізації федингу), тобто завмирання вводяться одночасно як в I, так і в Q- канали модулятора за однаковим алгоритмом. В термінах Matlab це реалізується наступною командою:

```
%*****Fading channel*****
[ifade,qfade]=sefade(ich2,qch2,itau,...
dlvl,th1,n0,itnd1,nowl,length(ich2),tstp,fd,flat);
```

Наступним кроком після введення завмирань є оновлення значення індексу завмирань:

```
% Update fading counter
itnd1=itnd1+itnd0;
```

В результаті вхідні дані у демодуляторі приймального тракту (в основній смузі частот) з урахуванням ефекту федингу і дії білого гаусівського шуму можна подати як:

```
%*****Add White Gaussian noise*****
ich3,qch3=comb(ifade,qfade, attn);
```

Далі сигнали у кожному каналі підлягають обробці у формуючому фільтрі приймача з наступним вирішенням про прийнятий біт у пороговому пристрої подібно до чистого методу QPSK (програма qpsk.m). Алгоритм визначення залежності BER від

відношення E_b/N_0 не змінюється, за єдиною різницею - на графік додатково виводиться залежність BER, яка характеризує криву релєївського порогу для методу QPSK, тобто, за умови проявлення тільки частотно-неселективних і повільних замирань. Організація циклу для значень E_b/N_0 повністю тотожна алгоритму, який використовувався в попередніх лабораторних роботах.

Теоретичний вираз для ймовірності помилкового прийому одного біта або коефіцієнта бітової помилки (BER) за наявності частотно-неселективних і повільних замирань для методу QPSK має такий вигляд: [3, 5]:

$$BER_{QPSK-FADING} = \frac{1}{2} \left[1 - \frac{1}{\sqrt{1 + \frac{1}{E_b/N_0}}} \right]. \quad (3.1)$$

Також для порівняння з AWGN каналом передбачено вивід теоретичної кривої залежності $BER=F(E_b/N_0)$ для AWGN каналу.

Відповідно блок виводу результатів моделювання набуває такого вигляду:

```
% *****Output result *****
ber=noe/nod;
noe=0;
nod=0;
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)=0.5*(1-1./sqrt(1+1/(10^(ebn0_array(ebn0_loop)/10))));
end

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

За таким алгоритмом виводяться графічні залежності, які отримані в результаті моделювання за запропонованим алгоритмом реалізації модему (масив **ber_array** – позначено як 'o'), теоретичним

виразом (3.1) для методу QPSK і наявності частотно-неселективних завмирань (масив **array_theory1** – позначено як '*') і теоретичним виразом (2.1) для методу QPSK при реалізації AWGN каналу (масив **array_theory** – позначено як '--').

3.2.2 Моделювання каналу зв'язку з використанням QPSK за наявністю частотно-селективних або швидких завмирань

Моделювання каналу зв'язку в умовах частотно-селективних завмирань реалізується за допомогою програми **qpsk_relay.m**.

Зміни у програмі порівняно з програмою моделювання повільних завмирань **qpsk_fading.m** підлягатимуть у наступному.

Для часу надходження відповідної затриманої хвилі (відносно прямої) вводиться нормалізований до кроку моделювання вектор **itau**:
itau = [0 floor(1*10.^(-6)/tstp) floor(1.5*10.^(-6)/tstp) floor(2.0*10.^(-6)/tstp)] = [0, 2, 3, 4];

Зменшення потужності затриманих хвиль відобразиться у векторі **dlvl**: **dlvl = [0,10,20,25];** .

Початкову фазу для усіх затриманих хвиль приймемо рівною нулю (вектор **th1**). Кількість хвиль для кожного шляху розповсюдження, як згадувалося, не повинна бути меншою 6. Це відобразимо у векторі **no**: **n0 = [6, 7, 6, 7];**

Кількість хвиль, що надійшла до приймача, відображається змінною **nowl = 4**. Доплерівське розсіювання задамо таким же, як і в ЛР №6 - **fd = 200** Гц. Відповідно змінна **flat** у випадку частотно-селективних завмирань набуває нульового значення.

Остаточно блок ініціалізації параметрів федингу у варіанті моделювання релеївських частотно-селективних завмирань приймає наступний вигляд:

```
%*****Fading initialization*****
tstp =1/sr/IPOINT;
itau=[0,2,3,4];
dlvl=[0,10,20,25];;
th1=[0.0, 0.0, 0.0, 0.0];
n0 = [6, 7, 6, 7];
itnd0=nd*IPOINT*100;
itnd1=[1000, 2000, 3000, 4000,];
nowl =4;
fd =200;
flat=0;
```

%***End of fading initialization*******

В іншому алгоритм моделювання, задіяні підпрограми і блок виводу результатів повністю тотожний програмі `qpsk_fading`.

3.3 Порядок проведення лабораторної роботи

3.3.1 Відкрити середовище Matlab. Створити нову директорію (папку) LAB9, в яку необхідно розмістити основні програми, а також усі підпрограми, що необхідні для моделювання QPSK сигналу в умовах релієвських завмирань.

3.3.2 Скопіювати файл основної програми `qpsk_fading.m`, який знаходиться у додатку В методичних вказівок і створити новий М-файл у директорії LAB9 під тим же ім'ям.

3.3.3 Скопіювати файл другої основної програми `qpsk_Relay.m`, який знаходиться у додатку В методичних вказівок і створити новий М-файл у директорії LAB9 під тим же ім'ям.

3.3.4 Повторити процедуру створення М-файлів в директорії LAB9 для усіх підпрограм, які відповідають алгоритмам реалізації моделі модему QPSK (усі підпрограми, які використовувалися у ЛР №8):

hrollcoef.m – коефіцієнти імпульсної характеристики фільтра Найквіста (корінь квадратний);

qpskmod.m – формування канальних символів для методу QPSK;

compoversamp.m – передискретизація в обох каналах I, Q модулятора;

compconv.m – операція згортки в обох каналах I, Q модулятора;

qpskdemod.m – алгоритм роботи порогових пристроїв;

comb.m – додавання гаусівського шуму до обох каналів I, Q демодулятора;

3.3.5 Повторити процедуру створення М-файлів в директорії LAB9 для усіх підпрограм, які відповідають алгоритмам генерації релієвських завмирань (усі додаткові підпрограми, які використовувалися у ЛР №6, ЛР №7):

fade.m – генерація доплерівського розсіювання;

delay.m – введення часу затримки для перевідбитих хвиль;

sefade.m – формування частотно-селективних завмирань.

Після виконання вище визначених пунктів в основному вікні середовища Matlab усі 11 М-файлів повинні відобразитися у полі поточної директорії LAB9.

3.3.6 У командній строчці набрати назву основної програми **qpsk_fading.m** і запустити аналіз натисканням клавіші Enter.

3.3.7 Отримати графічні залежності ймовірності помилкового прийому біта (або коефіцієнта BER) від відношення E_b/N_0 за умови наявності частотно-неселективних завмирань. Графічні залежності подати у напівлогарифмічному масштабі з діапазоном зміни ймовірності від 10^{-5} до 1. Діапазон зміни відношення E_b/N_0 задати від 0 до 35 дБ. Зафіксувати графіки.

3.3.8 У командній строчці набрати назву основної програми **qpsk_Relay.m** і запустити аналіз натисканням клавіші Enter.

3.3.9 Отримати графічні залежності ймовірності помилкового прийому біта (або коефіцієнта BER) від відношення E_b/N_0 за умови наявності частотно-селективних завмирань. Графічні залежності подати у напівлогарифмічному масштабі з діапазоном зміни ймовірності від 10^{-5} до 1. Діапазон зміни відношення E_b/N_0 задати від 0 до 35 дБ. Зафіксувати графіки.

3.3.10 Порівняти залежності BER для каналу з релеївськими завмираннями при використанні методів BPSK і QPSK.

3.4 Зміст звіту

3.4.1 Структурні схеми модуляторів і демодуляторів, що досліджуються. Блок-схема алгоритму програми.

3.4.2 Графічні залежності $BER=f(E_b/N_0)$ за результатами моделювання та теоретичними виразами в умовах частотно-неселективних завмирань.

3.4.3 Графічні залежності $BER=f(E_b/N_0)$ за результатами моделювання та теоретичними виразами в умовах частотно-селективних завмирань.

3.4.4 Лістинги програм.

3.4.5 Висновки за результатами роботи.

3.5 Контрольні запитання

1. Умови виникнення частотно-селективних завмирань у частотній області.
2. Умови виникнення частотно-селективних завмирань у часовій області.
3. Умови виникнення частотно-неселективних завмирань у часовій області.
4. Умови виникнення частотно-неселективних завмирань у частотній області.
5. Як програмно вводиться доплерівське розсіювання до моделі модемів.
6. Як програмно вводяться частотно-селективні завмирання до моделі модемів.
7. Пояснити програмний блок виводів результатів аналізу.
8. Методи і засоби боротьби з релеївськими завмираннями у стандарті GSM.
9. Методи і засоби боротьби з релеївськими завмираннями у стандарті cdmaOne.

ПЕРЕЛІК ПОСИЛАНЬ

1. Скляр Б. Цифровая связь. Теоретические основы и практическое применение [Текст] / Б. Скляр; Пер. с англ. ; -М. : Издательский дом “Вильямс”, 2003. – 1104с.
2. Ипатов В.П. Системы мобильной связи: Учебное пособие для вузов. [Текст] / В.П. Ипатов, В.К. Орлов, И.М. Самойлов, В.Н. Смирнов; под ред. В.П. Ипатова. -М. : Горячая линия–Телеком, 2003. – 272с.
3. Harada H. Simulation and software radio for mobile telecommunications [Текст] / H. Harada, R. Prasad; - Artech House, 2003.- 465 p.
4. Кабак В.С., Уваров Р.В. Функціональні пристрої телефонів мобільного зв'язку: Навчальний посібник. [Текст] / В.С. Кабак, Р.В. Уваров– Запоріжжя, 2007. – 375 с.
5. Феер К. Беспроводная цифровая связь. [Текст] / К. Феер; Пер. с англ. ; -М. : Радио и связь, 2000. – 519с.

ДОДАТОК А

Лістинг програми `bpsk_Relay.m`

```

%bpsk_Relay.m
%*****Preparation part*****
sr=256000.0; % Symbol rate
ml=1;
br=sr.*ml; % Bit rate
nd=100; % Number of bits in sequence
ebn0=10;
IPOINT=8; % Oversampling order
%*****Filter inialization*****
irfn=21;
alfs=0.5;
[xh]=hrollfcoef(irfn, IPOINT, sr, alfs, 1);
[xh2]=hrollfcoef(irfn, IPOINT, sr, alfs,0); % Receiver filter coefficients
%*****Fading initialization*****
tstp=1/sr/IPOINT; % time resolution
itau=[0,2,3,4]; % set three delayed waves
dlvl=[0,10,20,25];
n0=[6,7,6,7]; % number of waves
th1=[0.0,0.0,0.0,0.0]; % initial phase of delayed waves
itnd0=nd*IPOINT*100; % number of fading counter to skip
itnd1=[1000,2000,3000,4000]; % four fading counter is needed
nowl=4; % number of direct waves+ number of delayed waves
fd=200; % maximum Doppler frequency
flat=0; % frequency fading
%***** Start calculations*****
nloop=1000;
noe=0;
nod=0;
ebn0_array=0:1:35;
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*****
        data=rand(1,nd*ml)>0.5;
        data1=2.*data-1; % NRZ
        [data2]=oversamp(data1,nd,IPOINT); % ovsampling
    end
end

```

```

data3=conv(data2,xh); % convolution
% *****Attenuation Calculations*****
spow=sum(data3.*data3)/nd;
attn=0.5*spow*sr/br*10.^(-ebn0/10);
attn=sqrt(attn);
% *****Fading channel*****
[ifade,qfade]=sefade(data3,zeros(1,length(data3)),itau,...
dlvl,th1,n0,itnd1,nowl,length(data3),tstp,fd,flat);
    itnd1=itnd1+itnd0; % Update fading counter
% *****Add White Gaussian noise*****
inoise=randn(1,length(data3)).*attn;
data4=ifade+inoise;
data5=conv(data4,xh2);
sampl=irfn*IPOINT+1;
data6=data5(sampl:8*nd+sampl-1);
    % *****BPSK Demodulation*****
    [demodata]=data6>0;
    % *****Bit Error Rate*****
    noe2=sum(abs(data-demodata));
nod2=length(data);
noe=noe+noe2;
nod=nod+nod2;
fprintf('%d\t%e\n', iii, noe2/nod2);
end
% *****Output Results*****
ber=noe/nod;
noe=0;
nod=0;
    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)=0.5*(1-1./(sqrt(1+1/(10^(ebn0_array(ebn0_loop)/10)))));
end
fprintf('Eb/N0=%d\t BER=%e\t NoE=%d\t NoD=%e\n', ebn0, ber, noe, nod);
fid=fopen('BERbpsk.dat','a');
fprintf(fid,'Eb/N0=%d\t BER=%e\t NoE=%f\t NoD=%d\t\n', ebn0, ber, noe, nod);
fclose(fid);
figure;
h=semilogy(ebn0_array,ber_array,'o',ebn0_array,ber_array_theory,'--',...
ebn0_array,ber_array_theory1,'*');
set(h,{'DisplayName'},{'Experiment','Theory','Theory1'});
legend show;
% *****End of file*****

```

ДОДАТОК Б

Лістинги програм для моделювання каналу зв'язку з використанням методу QPSK

```

% qpsk.m
% Simulation program to realize QPSK transmission system
% *****Preparation part*****
sr=256000.0;    % Symbol rate
ml=2;          % ml : Number of modulation levels
br=sr.*ml;     % bit rate
nd=1000;       % Number of symbols
ebn0=10;       % 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);
    % Tranamitter filter coefficients
[xh2]=hrollfcoef(irfn,IPOINT,sr,alfs,0);
    % Receiver filter coefficients
% *****Start calculations*****
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; % rand: built in function
% *****QPSK Modulation*****
[ich,qch]=qpskmod(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;
    % sum: built in function
attn=0.5*spow*sr/br*10.^(-ebn0/10);
attn=sqrt(attn); % sqrt: built in function
% *****Add White Gaussian Noise (AWGN)*****
[ich3,qch3]=comb(ich2,qch2,attn); % add white Gaussian noise
[ich4,qch4]=compconv(ich3,qch3,xh2);
syncpoint=irfn*IPOINT+1;
ich5=ich4(syncpoint:IPOINT:length(ich4));

```

```

qch5=qch4(syncpoint:IPOINT:length(qch4));
%*****QPSK Demodulation*****
[demodata]=qpskdemod(ich5,qch5,1,nd,ml);
%*****Bit Error Rate (BER)*****
noe2=sum(abs(data1-demodata)); % sum: built in function
nod2=length(data1); % length: built in function
noe=noe+noe2;
nod=nod+nod2;
fprintf('%d\t%e\n',iii,noe2/nod2); % fprintf: built in function
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('BERqpsk.dat','a');
fprintf(fid,'%d\t%e\t%f\t%f\t\n',ebn0,noe/nod,noe,nod);
fclose(fid);
%*****End of file*****
figure; stem(0:40,data1(1:41),'DisplayName','data1');
figure; stem(0:20,ich(1:21),'DisplayName','ich');
figure; stem(0:20,qch(1:21),'DisplayName','qch');
figure;stem(0:20*IPOINT,ich1(1:20*IPOINT+1),'DisplayName','ich1');
figure; stem(0:20*IPOINT,qch1(1:20*IPOINT+1),'DisplayName','qch1');
figure;stem(0:20*IPOINT,ich2(1:20*IPOINT+1),'DisplayName','ich2');
figure; stem(0:20*IPOINT,qch2(1:20*IPOINT+1),'DisplayName','qch2');
figure; plot(0:20*IPOINT,ich2(1:20*IPOINT+1),'DisplayName','ich2');
figure; plot(0:20*IPOINT,qch2(1:20*IPOINT+1),'DisplayName','qch2');
figure; plot(0:20*IPOINT,ich3(1:20*IPOINT+1),'DisplayName','ich3');
figure; plot(0:20*IPOINT,qch3(1:20*IPOINT+1),'DisplayName','qch3');
figure; plot(0:20*IPOINT,ich3(1:20*IPOINT+1),'DisplayName','ich2');
figure; plot(0:20*IPOINT,qch3(1:20*IPOINT+1),'DisplayName','qch2');
figure; stem(0:20,ich5(1:21),'DisplayName','ich5');
figure; stem(0:20,qch5(1:21),'DisplayName','qch5');
figure; stem(0:40,demodata(1:41),'DisplayName','demodata');
figure; plot(ich5,qch5,'DisplayName','phaser');
%*****End of file *****

% qpskmod.m
% Function to perform QPSK modulation
function [iout,qout]=qpskmod(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)
% *****
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
iout((1:para),jj)=isi;
qout((1:para),jj)=isq;
count2=count2+m1;
end
% *****End of file *****

```

% qpskdemod.m

% Function to perform QPSK demodulation

```

function [demodata]=qpskdemod(idata, qdata, para, nd, ml)
% *****Variables *****
% idata : input Ich data
% qdata : input Qch data
% demodata : demodulated data (para by nd matrix)
% para : Number of parallel channels
% nd : Number of data
% ml : number of modulation levels
% ( QPSK -2, 16QAM -4)
demodata=zeros(para,ml*nd);
demodata((1:para),(1:ml:ml*nd-1))=idata((1:para),(1:nd))>=0;
demodata((1:para),(2:ml:ml*nd))=qdata((1:para),(1:nd))>=0;
% *****End of file *****

```

% compoversamp

```

% insert zero data to input data
function [iout,qout]=compoersamp(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 *****

```

% comconv.m

```

% Function to perform convolution between signal and filter
function [iout, qout]=comconv(idata,qdata,filter)
% idata :indata : ich data sequence
% qdata :indata : qch data sequence
% filter tap coefficients
% sample : number of oversamples
% *****
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 *****
% idata : input Ich data
% qdata : input Qch data
% 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 *****

%qpsk_ber.m
% Simulation program to realize QPSK transmission system
% *****Preparation part*****
sr=256000.0; % Symbol rate
ml=2; % ml : Number of modulation levels
br=sr.*ml; % bit rate
nd=1000; % Number of symbols
ebn0=10; % 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 calculations*****
nloop=100; % Number of simulation loops
noe=0; % Number of error data
nod=0; % Number of transmitted data
ebn0_array=0:1:15;
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; % rand: built in function
% *****QPSK Modulation*****
[ich,qch]=qpskmod(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; % sum: built in function
attn=0.5*spow*sr/br*10.^(-ebn0/10);
attn=sqrt(attn); % sqrt: built in function
% *****Add White Gaussian Noise (AWGN)*****
[ich3,qch3]=comb(ich2,qch2,attn); % add white Gaussian noise

```

```

[ich4,qch4]=compconv(ich3,qch3,xh2);
syncpoint=irfn*IPOINT+1;
ich5=ich4(syncpoint:IPOINT:length(ich4));
qch5=qch4(syncpoint:IPOINT:length(qch4));
%*****QPSK Demodulation*****
[demodata]=qpskdemod(ich5,qch5,1,nd,ml);
%*****Bit Error Rate (BER)*****
noe2=sum(abs(data1-demodata)); % sum: built in function
nod2=length(data1); % length: built in function
noe=noe+noe2;
nod=nod+nod2;
fprintf('%d\t%e\n',iii,noe2/nod2);% fprintf: built in function
end % for iii=1:nloop
%*****Output result *****
ber=noe/nod;
noe=0;
nod=0;
ber_array(ebn0_loop)=ber;
ber_array_theory(ebn0_loop)=0.5*erfc(sqrt(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,'-');
set(h,{ 'DisplayName','DisplayName'},{ 'Experiment','Theory'})
legend show
figure; plot(ich5,qch5,'DisplayName','phaser');
% *****End of file*****

```

ДОДАТОК В

Лістинги програм для моделювання каналу зв'язку з використанням методу QPSK і наявністю частотно-неселективних і частотно-селективних завмирань

```

%qpsk_fading.m
% Simulation program to realize QPSK transmission system
% *****Preparation part*****
sr=256000.0;    % Symbol rate
ml=2;          % ml : Number of modulation levels
br=sr.*ml;     % bit rate
nd=1000;       % Number of symbols
ebn0=10;       % 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);
    % Tranamitter filter coefficients
[xh2]=hrollfcoef(irfn,IPOINT,sr,alfs,0);
    % Receiver filter coefficients
% *****Fading initialization*****
tstp=1/sr/IPOINT; % time resolution
itau=[0]; % set only direct wave
dlvl=[0];
n0=[6]; % number of waves
th1=[0.0]; % initial phase of delayed waves
itnd0=nd*IPOINT*100; % number of fading counter to skip
itnd1=[1000]; % one fading counter is needed
nowl=1; % number of direct waves+ number of delayed waves
fd=200; % maximum Doppler frequency
flat=1; % flat fading
% *****Start calculations*****
nloop=200; % Number of simulation loops
noe=0; % Number of error data
nod=0; % Number of transmitted data
ebn0_array=0:1:35;
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; % rand: built in function
%*****QPSK Modulation*****
[ich,qch]=qpskmod(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; % sum: built in function
attn=0.5*spow*sr/br*10.^(-ebn0/10);
attn=sqrt(attn); % sqrt: built in function
%*****Fading Channel*****
% Generated data are fed into a fading simulator
[ifade,qfade]=sefade(ich2,qch2,itau,dlv1,th1,n0,...
itnd1,nowl,length(ich2),tstp,fd,flat);
itnd1=itnd1+itnd0; % Updating fading counter
%*****Add White GaussianNoise (AWGN)*****
[ich3,qch3]=comb(ifade,qfade,attn); % add white Gaussian noise
[ich4,qch4]=compconv(ich3,qch3,xh2);
syncpoint=irfn*IPOINT+1;
ich5=ich4(syncpoint:IPOINT:length(ich4));
qch5=qch4(syncpoint:IPOINT:length(qch4));
%*****QPSK Demodulation*****
[demodata]=qpskdemod(ich5,qch5,1,nd,ml);
%*****Bit Error Rate (BER)*****
noe2=sum(abs(data1-demodata));
% sum: built in function
nod2=length(data1); % length: built in function
noe=noe+noe2;
nod=nod+nod2;
fprintf('%d\t\t%e\n',iii,noe2/nod2); % fprintf: built in function
    end % for iii=1:nloop
%*****Output result*****
ber=noe/nod;
noe=0;
nod=0;
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)=0.5*(1-1./sqrt(1+1/(10^(ebn0_array(ebn0_loop)/10))));

```

```

end
fprintf('%d\t%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; plot(ich5,qch5,'DisplayName','phaser');
figure;
h=semilogy(ebn0_array,ber_array,'o',ebn0_array,ber_array_theory,'--
',ebn0_array,ber_array_theory1,'*');
set(h,{'DisplayName'},{'Experiment','Theory'})
legend show
% *****End of file*****

%qpsk_Relay.m
% *****Preparation part*****
sr=256000.0; % Symbol rate
ml=2; % ml : Number of modulation levels
br=sr.*ml; % bit rate
nd=1000; % Number of symbols
ebn0=10; % 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
% *****Fading initialization*****
tstp=1/sr/IPOINT; % time resolution
itau=[0,2,3,4]; % set three delayed waves
dlvl=[0,10,20,25];
n0=[6,7,6,7]; % number of waves
th1=[0.0,0.0,0.0,0.0]; % initial phase of delayed waves
itnd=nd*IPOINT*100; % number of fading counter to skip
itnd1=[1000,2000,3000,4000]; % four fading counter is needed
nowl=4; % number of direct waves+ number of delayed waves
fd=200; % maximum Doppler frequency
flat=0; % flat fading
% *****Start calculations*****
nloop=200; % Number of simulation loops
noe=0; % Number of error data

```

```

nod=0;    %Number of transmitted data
ebn0_array=0:1:35;
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; % rand: built in function
            %*****QPSK Modulation*****
            [ich,qch]=qpskmod(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;
            % sum: built in function
            attn=0.5*spow*sr/br*10.^(-ebn0/10);
            attn=sqrt(attn); % sqrt: built in function
            %*****Fading Channel*****
            % Generated data are fed into a fading simulator
            [ifade,qfade]=sefade(ich2,qch2,itau,dlv1,th1,n0,...
            itnd1,nowl,length(ich2),tstp,fd,flat);
            itnd1=itnd1+itnd0; % Updating fading counter
            %*****Add White GaussianNoise (AWGN)*****
            [ich3,qch3]=comb(ifade,qfade,attn); % add white Gaussian noise
            [ich4,qch4]=compconv(ich3,qch3,xh2);
            syncpoint=irfn*IPOINT+1;
            ich5=ich4(syncpoint:IPOINT:length(ich4));
            qch5=qch4(syncpoint:IPOINT:length(qch4));
            %*****QPSK Demodulation*****
            [demodata]=qpskdemod(ich5,qch5,1,nd,ml);
            %*****Bit Error Rate (BER)*****
            noe2=sum(abs(data1-demodata)); % sum: built in function
            nod2=length(data1); % length: built in function
            noe=noe+noe2;
            nod=nod+nod2;
            fprintf('%d\t%e\n',iii,noe2/nod2); % fprintf: built in function
        end % for iii=1:nloop
    %*****Output result *****
    ber=noe/nod;
    noe=0;

```

```

nod=0;
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)=0.5*(1-
1./(sqrt(1+1/(10^(ebn0_array(ebn0_loop)/10))));
end

fprintf('%d\t%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; plot(ich5,qch5,'DisplayName','phaser');
figure;
h=semilogy(ebn0_array,ber_array,'o',ebn0_array,ber_array_theory,...
'--',ber_array_theory1,'*');
set(h,{'DisplayName'},{'Experiment','Theory'})
legend show

% *****End of file*****

```